Developing a smart parking solution based on a Holonic Multiagent System using JaCaMo Framework

Lucas Fernando Souza de Castro[1], André Pinz Borges[1], Gleifer Vaz Alves[1]

[1]Programa de Pós-Graduação em Ciência da Computação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa (UTFPR)
Av Monteiro Lobato, s/n - Km 04 - Ponta Grossa - PR - Brazil

lcastro@alunos.utfpr.edu.br,{gleifer,apborges}@utfpr.edu.br

Abstract. *A smart city is described as an intelligent city composed of technological services to improve the human life in a way to make it easier and practical. One of the most costly things to optimize in a smart city is the traffic, particularly, the process of looking for a parking space. Therefore, some computer solutions have been developed to provide a parking space an autonomous way in a smart city, and one of them is using Multiagent System (MAS). This paper proposes a Holonic Multiagent System (HMAS) to assign and manage parking spaces in a smart parking system called Holonic Multiagent Parking System (MAPS-HOLO) developed through JaCaMo Framework which comprises three layers of MAS programming: agent, environment and normative programming. Besides the assign parking space process, the system will be able to handle runtime agents failures in different levels: driver agent, sector agent, and manager agent failure.*

1. Introduction

A smart city is a city with technological systems able to make easier the daily routine of the population, such as: pay the municipal property tax through a mobile app, use a public wireless network walking on a park or even find a parking space before leaving home. Considering that a smart city there are many things could be optimized, [Caragliu et al. 2011] introduces six categories which compose a smart city: smart economy, smart mobility, smart environment, smart people, smart living, and smart governance.

The smart mobility comprises systems able to further easier access to move around the city (e.g., walking or driving). In New York, a motorist spends almost 40% of its driving time looking for a parking space [Koster et al. 2014]. Hence, some technological solutions have become common to find a parking space in some cities around the world. For example, cities like San Francisco (USA) [ParkingEdge 2013], Naples (Italy) [Napoli et al. 2014], and São Paulo [Dias and Este 2016] are examples of places where computer solutions provide an easier access to find/reserve a parking space.

The vast majority of research concerning smart parking technology is towards smart cities. Besides the smart city, a parking lot should have some te to be classified as a smart parking [Revathi and Dhulipala 2012]. Most of the technologies used by smart parking employ some intelligent systems, such as fuzzy logic, computer vision, and multiagent systems. These technologies have been used by different scenarios, however, in

[Fraifer and Fernström 2016] the authors presented the main trends in smart parking context. Figure 1 shows a chart of the trends in technology.

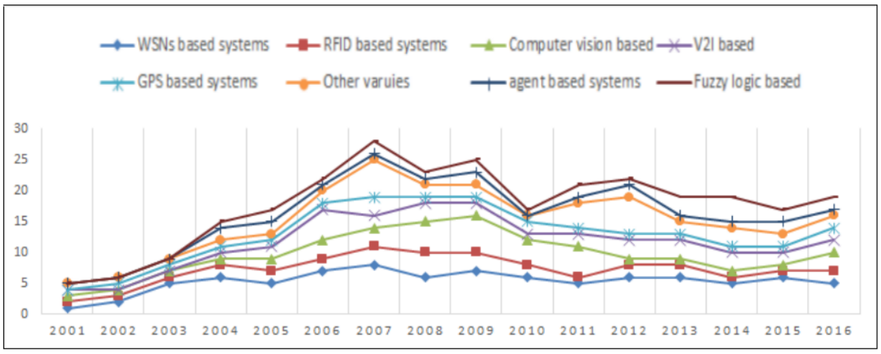

Figure 1. Smart parking technologies trends [Fraifer and Fernström 2016]

As seen in the Figure 1, the most used technology in the smart parking context is the fuzzy logic due to the usage of sensors and their requirement to calibrate and detect errors. Likewise, in the second place is the MAS due to the characteristics of a parking lot (dynamicity of drivers, usage/monitoring parking spaces, sensors, and payments [Idris et al. 2009]). The MAS is being used to build solutions to organize and manage smart parking. Moreover, it could be applied in monitoring a parking space through sensor or a camera [Basavaraju 2015], assign parking spaces and even simulate the environment.

Allocating parking spaces may be a complex task due to the limited amount of them. Hence, it is necessary to evaluate each parking request based on some pattern. One pattern which could be applied is the trust degree value, which is related to agent's commitment to the rules and usage of the system. The trust degree was applied in a MAS through JaCaMo Framework[1], which is a framework with Jason[2] (agent programming), Cartago[3] (artifact development), and Moise[4] to the normative programming [Boissier et al. 2013] to allocate parking spaces, where the driver agents compete with each other to win a parking space given by a manager agent based on their trust degree value [Castro et al. 2017].

In this paper, we propose a new MAS able to assign and manage smart parking, which indeed is based on a Holonic Multiagent System (HMAS), i.e., a MAS with holons or holonic agents. The term HOLONIC was designed by Arthur Koester in [Koestler 1967] where he showed how the universe and human body could be observed holons. The HOLON is a Greek word which means whole and part simultaneously. Then, this concept applied to Koster's idea results that universe is composed of a whole (universe itself) and its parts (planets, galaxies, etc.) and so on. A holonic agent is an agent capable of having nested agents and to create holarchies (grouping of holonic agents) with other holonic agents [Giret and Botti 2004]. Inside of a holarchy, it is necessary an organizational model govern the agents. A conventional model is the head-body, where a head agent is acting as a leader and the body agents working as cooperators [Fischer et al. 2003].

[1] JaCaMo Website: http://jacamo.sourceforge.net/
[2] Jason: http://jason.sourceforge.net/wp
[3] Cartago: http://cartago.sourceforge.net/
[4] Moise: http://moise.sourceforge.net/

Our HMAS is divided into four categories of agents: *manager, sector, parking space, and driver Agent*. The primary goal of this paper is to propose the development of a system able to assign parking spaces to *Driver Agents* and deal with runtime agent failures. The failures might happen in each agent. However, every failure has to be handled differently according to the agent (*manager, sector, parking space,* and *driver*). The development of the HMAS will be on JaCaMo Framework. Moreover, one of the goals of `MAPS-HOLO` is to evaluate the usage of JaCaMo to develop HMAS.

Furthermore, the usage of holonic agents in the MAS applied in a smart parking scenario may have the following advantages: (i) a higher capacity to deal with the failures due to the usage of holarchies; (ii) higher scalability due to the concept of HOLOS, where the HMAS can run as a whole (a group of holarchies) or/and run as part (single holon).

The remainder of this paper is organized as follows. In Section 2 is described the architecture of the HMAS and how we intend to handle the runtime agent failures. Finally, Section 3 presents the final considerations.

2. Proposal of Holonic Multiagent System

The proposed paper is part of a research initiative named MultiAgent Parking System (MAPS), where some systems have been conducted through JaCaMo Framework to provide smart parking. In this section, we present the architecture designed for the `MAPS-HOLO` as well as discuss some possible agent's failures and how we intend to solve them.

2.1. `MAPS-HOLO` Architecture: General Overview

The architecture of the `MAPS-HOLO` is divided into four types of agents: *manager, sector, parking space, and driver agent*. The usage of different types of agents to manage the smart parking provides a lighter workload to the agents due to the task sharing among agents (e.g., parking space monitoring). As it follows, Figure 2 shows a holonic perspective of the system compared with the JaCaMo Framework layers (agent, artifacts, and organizational layer). The diagram *A* (on the left) presents the holonic representation with the holons and their interactions, whereas the diagram *B* shows the holons mapped to the JaCaMo perspective.

As seen in the Figure 2, every agent has its own features and duties as follows:

- **Manager Agent:** It receives the parking spaces requests from drivers and validates the inquiries based on their preferences (parking space location, and price). After the validation process, it sends to the *sector agent* the parking space request. Finally, after the answer from *sector*, the agent sends to the driver the result of the request (Success if the driver received a parking space according to its preferences, and failure if the driver did not receive the parking space);
- **Sector Agent:** The *sector agent* is in charge to manage the parking spaces in its sector. After it receives the request from the *manager*, the agent chooses an available parking space and send the request to the *parking space agent*. If the *parking space agent* refuses the request, the *sector agent* has to choose another agent and repeat the process;
- **Parking Space Agent:** It is in charge to observe the physical parking space through sensors. The agent may have three different status: busy (if there is a

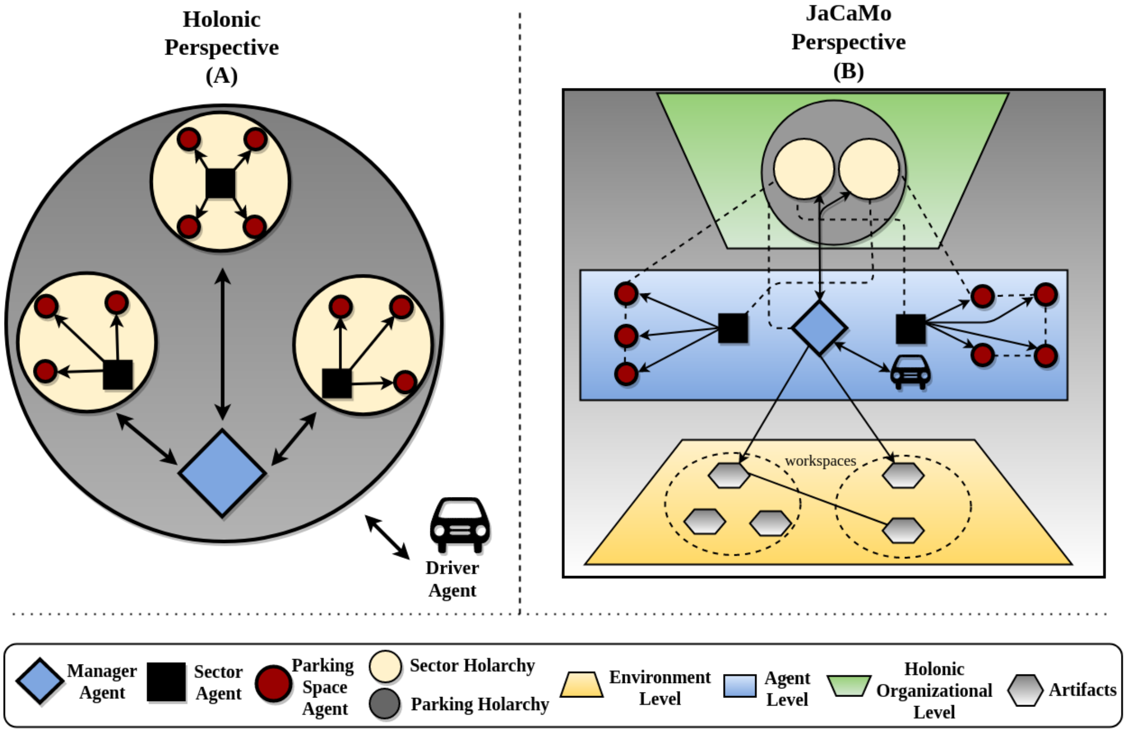

Figure 2. MAPS-HOLO Holonic/JaCaMo - Perspective

driver agent using it), free (available to be assigned) or reserved (when the parking space was assigned to a *driver agent*, but it did not park yet);

- **Driver Agent:**It represents the actual vehicle which needs a parking space. The drivers need to set its preferences (location and price) to request a space. Then, the driver will receive the answer to its request, and it may accept or refuse it.

Although the holarchies of the HMAS could be seen as agent organizations, the holarchy in the MAPS-HOLO will provide more features than a simple organization of agents. Every holarchy owns a contract, which a group of commitments that the agents have to follow [Fischer et al. 2003]. This contract prescribes how the agents interact (e.g., direct messages or broadcasting), rules of resource sharing, and the strategies about how the holarchy handles the failures. The JaCaMo through Moise provides development of organizations with roles and norms to the agents which make viable to create standard holarchies. There are two holarchies (as shown in Figure 2) in the HMAS: *Parking Holarchy*, and *Sector Holarchy*.

- **Parking Holarchy:** It is composed by three groups of agents: *manager, sector and parking space*. Its head agent is the *manager* who is responsible to coordinate the smart parking environment, and the body agents are the *sector* and *parking space agents*. It has a contract, which is a set of commitments that regulates the holarchy such as: i) Communication among *manager* and *sectors agents*; ii) Range of *sector agents*; iii) Order and priority among *sectors*;
- **Sector Holarchy:** The *Sector Agent* is its head agent, and the body agents are the parking spaces. Its contract specifies how the *parking space agents* perceive the physical parking space (e.g., interval to check sensors) and how the body agents interact with each other.

2.2. Agent Failures

One advantage of use holons is how the agents get connected with the others through holarchies providing a higher capacity to face runtime agent failures that could finish a whole MAS execution. However, the simple usage of holonic agents does not solve the agent failures automatically. Hence, it is necessary to program how the HMAS will handle the failures and check if there is an agent failure. To check if a failure occurred, periodically the agents must inform an artifact with their state. If an agent reached the timeout, the *observer agent* is notified and will handle the failure. The following topics describe how the failures will be dealt with:

- **Parking Space Agent Failure:** The agent notifies the *Sector Agent* of its holarchy. The head agent of its holarchy picks another *parking space agent* to manage the resources from the failed agent;
- **Sector Agent Failure:** Due to the *sector agent* being the head agent of *sector holarchy*, the following steps will be executed: i) All the *parking space agents* of its holarchy will be notified; ii) Another *sector agent* will be notified by the *manager agent* to manage these *parking space agents* notified in the previous step by the *manager*; iii) The notified *sector agent* manages now the new *parking spaces agents*;
- **Manager Agent Failure:** As one may expect, the manager failure will represent many changes in the system in order to properly recovery it. All the *sector agents* will be notified about the *manager's* failure and the *Parking Holarchy* will have the head agents changed to multiple head agents (Sector Agents). Hence, the process of assigning/managing parking spaces will be changed to a self-organization and coordination of *sector agents*. Moreover, the *manager* has an important role when the *parking space* and/or *sector agent* fail. Then, to handle these failures a new type of agent will be used: *Observer*. Besides the failures, the *observer agent* will inspect the self-coordination among the *sector agents* to check how they are working. The *observer agent* does not operate when the system starts or when the *manager* runs, the *observer* is employed only when the *manager* fails.

Given, the attempts to handle the failures, we expect that the *driver agent* can use the system without interruptions even if the agents fail. Moreover, our proposal of using an HMAS is to provide a higher capacity to deal with failures and scalability to build a stable system to face the dynamicity of smart parking.

3. Final Considerations

The primary goal of our paper is to propose a smart parking solution based on Holonic Agents. The system will be able to assign and manage parking spaces in smart parking through holonic agents and their holarchies. The usage of holarchies will provide a higher capacity to deal with different types of failures (Manager, Sector, and Parking Space). Finally, whereas the `MAPS-HOLO` will be finished some contributions could be highlighted: i) HMAS to manage/assign parking spaces in a smart parking; ii) The usage of holonic agents through JaCaMo Framework; iii) Usage of `MAPS-HOLO` not only for cars, but also different sorts of resources, such as bike, computer time, etc.); iv) Plans to handle runtime agent failures. The next steps to build the `MAPS-HOLO` are: i) Agent and artifact programming; ii) Mapping holarchies through Moise; iii) Developing holarchies artifacts; iv) Submitting the HMAS to programmed and random failures; v) Implementing the `MAPS-HOLO` in a real scenario.

Acknowledgments

Thanks for the support of CAPES by means of the masters scholarship.

References

Basavaraju, S. R. (2015). Automatic smart parking system using internet of things (iot. *International Journal of Scientific and Research Publications*, 5(12):629–632.

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.

Caragliu, A., Bo, C. D., and Nijkamp, P. (2011). Smart cities in europe. *Journal of Urban Technology*, 18(2):65–82.

Castro, L. F. S., Alves, G. V., and Borges, A. P. (2017). Using trust degree for agents in order to assign spots in a smart parking. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):45.

Dias, E. M. and Este, W. (2016). Sistema Inteligente de Estacionamento Público Integrado-SIEPI. *Journal of Progressive Research in Mathematics*, 7(4):1173–1182.

Fischer, K., Schillo, M., and Siekmann, J. (2003). Holonic multiagent systems: A foundation for the organisation of multiagent systems. In *HoloMAS*, pages 71–80. Springer.

Fraifer, M. and Fernström, M. (2016). Investigation of Smart Parking Systems and their technologies. In *Thirty Seventh International Conference on Information Systems. IoT Smart City Challenges Applications (ISCA 2016), Dublin, Ireland*, pages 1–14.

Giret, A. and Botti, V. (2004). Holons and agents. *Journal of Intelligent Manufacturing*, 15(5):645–659.

Idris, M., Leng, Y., Tamil, E., Noor, N., and Razak, Z. (2009). Car Park System: A Review of Smart Parking System and its Technology. *Information Technology Journal*, 8(2):101–113.

Koestler, A. (1967). The ghost in the machine. 1967. *London: Hutchinson*.

Koster, A., Koch, F., and Bazzan, A. L. (2014). Incentivising crowdsourced parking solutions. In Nin, J. and Villatoro, D., editors, *Citizen in Sensor Networks*, volume 8313 of *Lecture Notes in Computer Science*, pages 36–43. Springer International Publishing.

Napoli, C. D., Nocera, D. D., and Rossi, S. (2014). Agent negotiation for different needs in smart parking allocation. In *Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection - 12th International Conference, PAAMS 2014, Salamanca, Spain, June 4-6, 2014. Proceedings*, pages 98–109.

ParkingEdge (2013). San francisco - best parking. available in: http://sanfrancisco.bestparking.com (accessed in 1st march 2018).

Revathi, G. and Dhulipala, V. (2012). Smart parking systems and sensors: A survey. In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, pages 1–5.