Development of Agent-Based CPS for Smart Parking Systems

Lucas Sakurada†, José Barbosa*, Paulo Leitão*, Gleifer Alves†, André Pinz Borges†, Pedro Botelho†
* Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança,
Campus de Santa Apolónia, 5300-253 Bragança, Portugal
Email: {jbarbosa, pleitao}@ipb.pt
† Federal University of Technology - Paraná (UTFPR)
Email: lucas-sakurada@hotmail.com, {gleifer, apborges}@utfpr.edu.br, pbotelho@alunos.utfpr.edu.br

***Abstract*—The increase volume of vehicles circulating in large cities and the limited space for parking are factors that motivate the adoption of systems capable of dealing with such problems. In this context, smart parking systems are suitable solutions to avoid the traffic congestion, the air pollution and the long search to find a free parking spot. The inclusion of emergent ICT technologies and artificial intelligence techniques, and particularly using multi-agent systems, combined under the scope of Cyber-Physical Systems (CPS), ensure flexibility, modularity, adaptability and the decentralization of intelligence through autonomous, cooperative and proactive entities. Such smart parking systems can be easily adapted to any type of vehicle to be parked and scalable in terms of the number of parking spots and drivers/vehicles. A fundamental issue in these agent-based CPS parking systems is the interconnection between the cyber and physical counterparts, i.e. between the software agents and the physical asset controllers to access the parking spots. This paper focuses on developing an agent-based CPS for a smart parking system and particularly addressing how the software agents are interconnected with the physical asset controllers using proper Internet of Things technologies. The proposed approach was implemented in two distinct parking systems, one for bicycles and another for cars, showing an efficient, modular, adaptable and scalable operation.**

Keywords: Cyber-Physical Systems, Multi-Agent Systems, Internet of Things, Smart parking.

I. INTRODUCTION

In the last decades, the urbanization process has become to be increasingly intense, being the main obstacle for the sustainable development of the large cities. Among the problems caused by this process, the incessant increase of the circulation of vehicles and the limited space are factors that only contribute to aggravate this situation. The simple search for a free parking spot may cause the traffic congestion and the frustration to the driver, consuming a significant amount of time and contributing for the air pollution [1], [2].

Currently the most common way to find a parking spot is through attempts, without the help of a decision-support system [1], [3], that can use the available real-time data related to the parking systems occupancy to suggest to drivers where to park their vehicles. Smart parking systems have the necessary requirements to deal with such problems, where the objective is to ensure that the driver can find a free parking spot for parking its vehicle in an efficient way (i.e. in the shortest time possible and as close as possible to the desired location), avoiding the traffic congestion and environmental pollution [1], [2], [3], [4]. In this context, the driver can dynamically and in real-time to start a negotiation with parking spots to reserve a parking spot or receive a parking offer from the parking spots.

As a complex and large-scale system, a smart parking system needs to adopt technologies that offer the requirements for its implementation, constituting a CPS. Although the CPS is closely related to industrial processes, its potential extends to other fields of application, such as traffic management [5]. CPS integrates a network of computational resources and physical processes, basically consisting of a physical part capable of interacting with the real world through, e.g., sensors and actuators, and a cyber part responsible for processing the obtained data [6]. CPS can be augmented with the use of Multi-Agent Systems (MAS) [7], which allows the incorporation of intelligence into the CPS, decentralizing the system through autonomous, proactive and cooperative entities, namely software agents. The agent-based configuration, characterized by modularity, flexibility, robustness and responsiveness, is a suitable approach for the design of complex and large-scale systems [7], [8], as smart parking systems are. In such CPS parking systems, besides the negotiation among the autonomous agents to allocate vehicles to parking spots, the interconnection between the software agents and the low-level control devices assumes a crucial importance. This interconnection can be performed by using several technologies and protocols, being necessary to guarantee that the adopted follows the parking system requirements, namely to be scalable and loosed coupled.

Having this in mind, this paper describes the development of an agent-based CPS for smart parking systems, focusing the interconnection between the software agents with the low-level control devices to control the access of vehicles to the parking spots, using proper interface practices based on Internet of Things (IoT) technologies and a publish-subscribe model, to address the established requirements. The designed agent-based CPS was implemented in two prototype parking systems, one specific for bicycles and another for cars, showing the properly, robust and scalable operation.

The rest of the paper is organized as follows. Section II overviews the agent-based CPS architecture for smart parking systems and Session III addresses in more detail the cyber-

978-1-7281-4878-6/19/$31.00 ©2019 IEEE

physical interconnection, proposing a generic and loosed coupled structure for the interaction between the software agents and the physical asset controllers that regulate the access of vehicles to the parking spots. Section IV presents the prototype implementation of the agent-based CPS for two parking systems, one for bicycles and another for cars, and Section V analyzes the achieved experimental results. Finally, Section VI rounds up the paper with the conclusions and points out the future work.

II. Agent-based CPS Architecture for Smart Parking Systems

The agent-based CPS architecture for smart parking systems is based on the use of MAS technology aligned with the holonic principles, as illustrated in Figure 1.

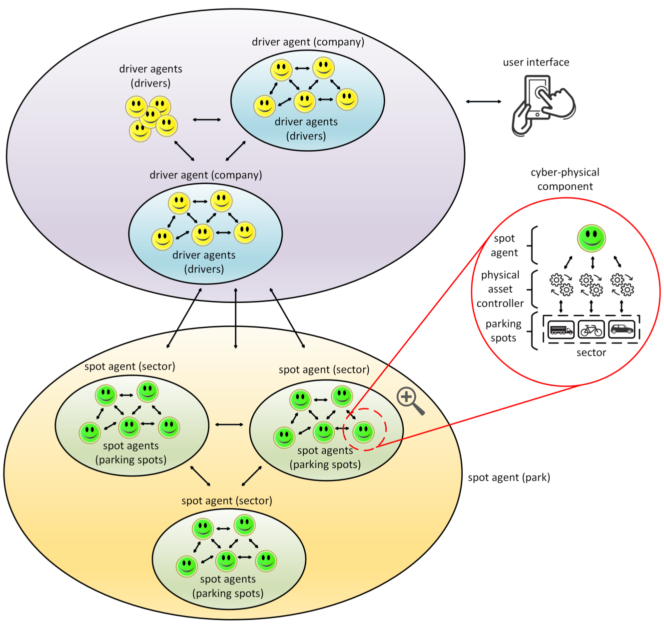

Figure 1: Holonic agent-based CPS smart parking system architecture.

This architecture uses autonomous and cooperative software agents to represents the several components of the parking systems, particularly defining two types of agents: spot and driver agents. The driver agents represent the drivers who want to park their vehicles and are running typically in the cloud, and the spot agents are responsible for managing the available parking spots, being possible to run in the cloud or edge computing levels, e.g., using powerful single-board computers. The agent-based system has as many driver agents as the number of drivers and as many spot agents as the number of available parking spots in the parking system. Also flexible is the possibility that each spot agent can manage several parking spots.

Each driver and spot agent has its own objectives and role in the system: the driver agents trying to get parking spots as cheap and close to the desired place as possible, and the spot agents trying to get as much reward as possible. For this purpose, they are executing a set of behaviours to achieve and optimize their objectives, and may embed artificial intelligence techniques to support the proper execution of these behaviours, mainly addressing the negotiation process. As examples, machine learning techniques can be used for the dynamic prediction of occupancy of parking spots, taking into account participatory decision variables and atmospheric forecasting variables. For example, a frequent user who prefers to use a certain parking spot due to its age or locomotive difficulties, should maintain an automatic reservation that favors him. In addition to these participatory variables in the decision, the atmospheric forecast variables, e.g., temperature and rainfall, should be taken into account in the correlation between the user and the system usage.

The adoption of MAS ensures inherent characteristics of this technology, such as flexibility, modularity and adaptability, to design smart parking systems. In fact, in such systems, the overall system operation emerges from the interaction among the distributed agents, each one contributing with its local knowledge and skills. The system can be easily adapted to any type of vehicle (e.g., a bike, car or truck), and re-configured on-the-fly, i.e. new parking spots or drivers can be easily added, removed or modified without stopping, reprogramming and restarting the entire system. These interactions can assume different patterns contributing to reach the system goals. Particularly, the negotiation process assumes a crucial interaction pattern in smart parking systems, where driver agents interact with spot agents to reserve a parking spot respecting the specifications defined by the driver, e.g., location, type and parking time. For this purpose, several negotiation strategies can be used, namely the Contract Net Protocol [9], the English auction [10] and the Dutch auction [11], which applicability to smart parking systems were tested and compared in [12].

The use of holonic principles simplifies the design of such large-scale smart parking systems, particularly considering the Janus effect of an holon, where an holon can be seen simultaneously as a whole and a part of the system [13]. In fact, considering the holonic's inherent recursivity property, a driver agent can be at the same time the "whole", e.g., representing a driving company, and the "part", e.g., representing a single driver, as shown in Figure 1. The same applies to a spot agent, which can represent a sector of parking spots comprising several parking spots or represent a single parking spot.

In this CPS architecture, the drivers use a web application running in the smartphone to access to the smart parking system, e.g., to reserve a parking spot, use a reserved parking spot or consult the historic of reservations. Particularly, the driver interconnects with the driver agent that acts on his behalf through the web application. Some other agents have physical representation that require the interconnection between the cyber and physical parts, namely the spot agent that constitutes the cyber part and interacts with the physical spot asset controller. Although an overview of the smart parking system functioning has been presented, the current work focuses on the low-level layer, i.e., on the interconnection of each spot agent with its respective physical asset controller.

III. Cyber-Physical Interconnection

The interconnection between the software agent and the low-level control device can be performed in several ways, namely considering client-server or publish-subscribe protocols. Based on the experimental tests reported in [8], which compares the application of different interface practices in terms of response time and scalability, the interface practice that better fits the smart parking system problem is the *Remote/Publish-Subscribe* [8], where the spot agent and the physical asset controller are executed in different computational platforms and interact using the publish/subscribe model that ensures the loosed coupled requirement. This approach provides the necessary requirements for the proper functioning of the system, where the real-time response is not critical and the scalability is much more significant for this application.

Taking into account the use of this interface practice for the interconnection between the spot agents and the physical asset controllers, the proposed generic modular structure for the cyber-physical interconnection is illustrated in Figure 2.

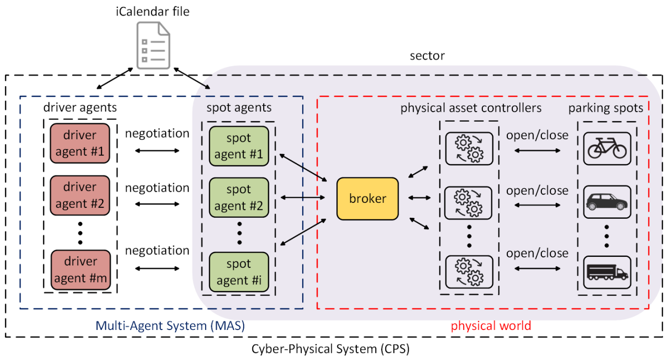

Figure 2: Structure for the cyber-physical interconnection.

The spot agents are responsible for managing a sector comprised of several parking spots that are controlled by physical asset controllers. On the other hand, the driver agents, representing the drivers that want to park their vehicles, initiate a negotiation with the spot agents to reserve the parking spot. Once the negotiation process is concluded, the reservation information is stored in a iCalendar file [14] and the parking spot is reserved to be used by the driver in the specific slot of time.

The iCalendar file enables the simple and efficient transfer of calendar and scheduling information between different computer systems. The file contains the reservation information for the parking system, including the following fields: driverId, parkingSpotId, status (i.e., if the parking spot is free, occupied or timeout), startParkingDate and EndParkingDate. As a smart parking is constantly updated with new reservations, the spot agent needs to read this file whenever it is requested the access to the parking spot.

Since the communication between the spot agent and the physical asset controllers is performed by message topics through a broker, the parking spot topic is required for the spot agent to know what message topic is necessary to interact with the physical asset controller that regulates the access to the reserved parking spot. Figure 3 illustrates how the spot agent and physical asset controllers interact through the use of message topics following the publish-subscribe model.

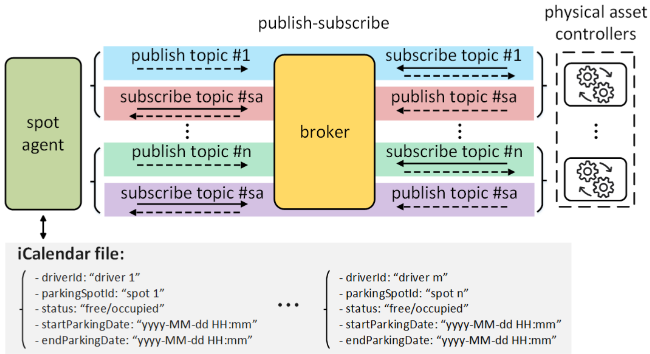

Figure 3: Using a broker to support the exchanging of messages following the publish-subscribe model.

The spot agent uses specific topics to communicate with each physical asset controller, maintaining an independent communication. In this case, any message published by the spot agent in a specific topic will be forward by the broker to the physical asset controllers that have subscribed such topic. The communication from the physical asset controllers to the spot agents follows the same structure.

The interaction sequence among the different system players, and particularly between the spot agent and the physical asset controller, during the use of the parking spot reservation, is illustrated in Figure 4.

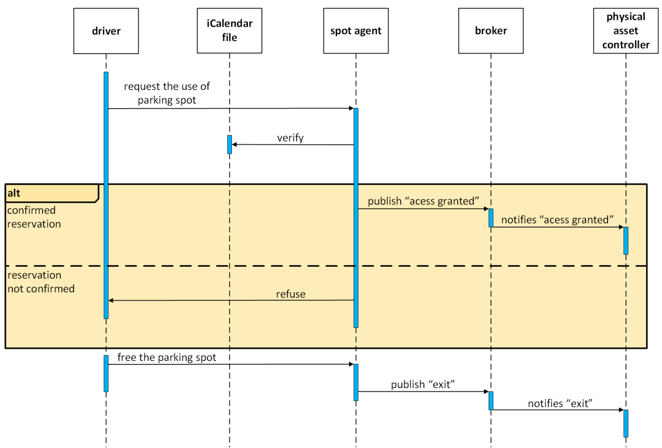

Figure 4: Interaction diagram for the cyber-physical interconnection.

Briefly, when the driver wants to park its vehicle, the driver agent sends a request to the spot agent responsible for managing the reserved parking spot, which checks in the iCalendar file if the reservation is valid. If it is correct, the spot agent sends an access command to the physical asset controller located in the reserved parking spot to allow the parking of the vehicle (e.g., opening the gate). If none reservation is found, or the start time has not yet started or the reservation time has already expired, the spot agent notifies the driver agent that can not access the parking spot. In the same manner, when the driver wants to exit the parking spot, the driver agent sends a request to the spot agent, which forwards to its respective physical asset controller to free the parking spot.

IV. Experimental Implementation

The proposed agent-based CPS for smart parking is deployed in two different types of parking systems: one for bicycles and another for cars. Both of CPS parking systems are included in environments with intense flow of people, such as universities, and are composed by sectors, each one comprising a spot agent, a broker, several physical asset controllers and several parking spots. The spot agent is responsible for managing the parking spots, where each one has a physical asset controller to allow the use of the parking spot according to the reservation previously stored in the iCalendar file.

The agent-based system for the parking system was developed using JADE (JAVA Agent Development Framework) [15], being the spot agents deployed in Raspberry Pi 3 Model B+ single-board computers. The communication between the spot agent and the physical asset controller is established via a broker implemented using the Eclipse Mosquitto [16] that implements the MQTT (Message Queuing Telemetry Transport) protocol.

A. Implementation of an Agent-Based CPS for Parking of Bicycles

The development of an agent-based CPS for a parking system of bicycles aims to facilitate the daily use of bicycle drivers, stimulate the practice of physical activity and promote the sustainable development by encouraging the use of bicycles rather than vehicles responsible for carbon dioxide emissions.

As illustrated in Figure 5, the spot agent controlling a sector of parking spots for bicycles was deployed in the Raspberry Pi #1. Each physical asset controller, managing the use of the parking spot (e.g., through a latch that will lock the bicycle in the parking spot and release after use), comprises a logical control that is running in the ESP8266 microcontroller, and is responsible for giving or not accessing to the bicycle parking spot. The iCalendar file is located in the Raspberry Pi #1 and the Eclipse Mosquitto is running in the Raspberry Pi #2.

Additionally, and aiming to test the scalability of the agent-based parking system, several virtual parking spots were created. For this purpose, 7 additional sectors were added, being each one managed by one spot agent and comprising 6 parking spots (each one controller by a logic control running in a virtual ESP8266).

Figure 6 shows the monitoring dashboard of the smart parking system developed in Node-RED, a visual flow-based programming tool that simplifies the development of IoT applications. This dashboard shows a map of the parking system and the current status of each parking spot: a red circle when the parking spot is occupied, a green circle when the parking spot is free and a black circle when the parking spot is occupied, but the reservation time has expired. In addition, the driver Id is shown to know which driver is using the parking spot.

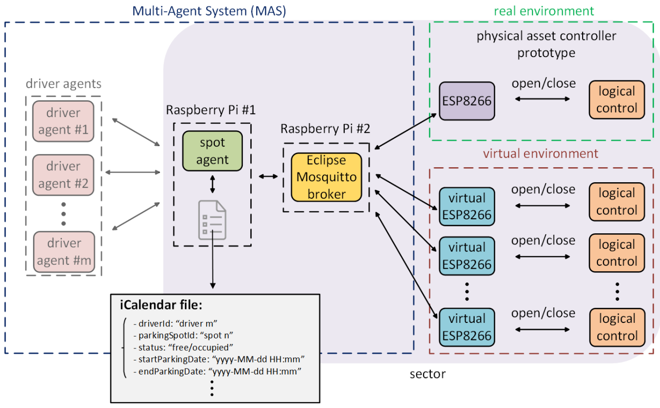

Figure 5: Implemented cyber-physical interconnection for the parking system of bicycles.

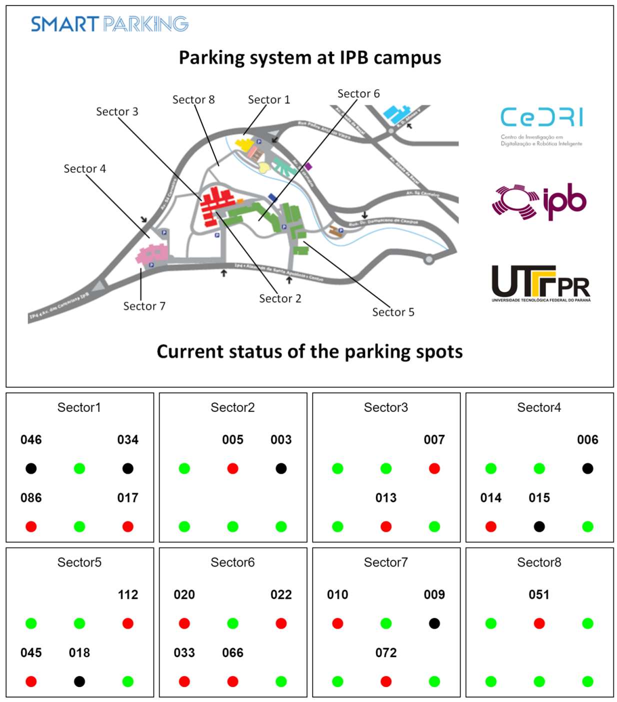

Figure 6: Monitoring dashboard for the bicycles smart parking.

In order to test the implemented cyber-physical interconnection for the bicycle parking, an experimental scenario was considered to verify the functioning of the system. Thus, some commands were requested for the spot agent of each sector, such as access or release the parking spot, checking through the Node-RED dashboard, the communication of the spot agent with their respective physical asset controllers to allow the access to the parking spots according to reservations in iCalendar file.

The virtual asset controllers are also running in the Node-RED platform, interacting with the spot agents through the exchange of messages via the Mosquitto broker.

B. Implementation of an Agent-Based CPS for Parking of Cars

The implementation of a CPS parking system for cars aims to manage the parking spots available to the drivers that want to park their cars. The parking of cars demands a different infrastructure when comparing to the parking of bikes, which requires some tweakings on the architecture in order to properly work. The parking of cars includes some sectors distributed in off-street parking, each one comprises several parking spots managed by the spot agent responsible for a given sector, as shown in Figure 2.

For the parking of cars, it is not feasible to have a lock or a gate that controls every parking spot for cars. In this way, and as illustrated in Fig. 7, the asset controller comprises the ESP8266 microcontroller, the ultrasonic sensor Model HC-SR04, and two LEDs that indicate the parking spot status in the physical layer: the red light indicates that the spot is busy, and the green light indicates that the spot is free.

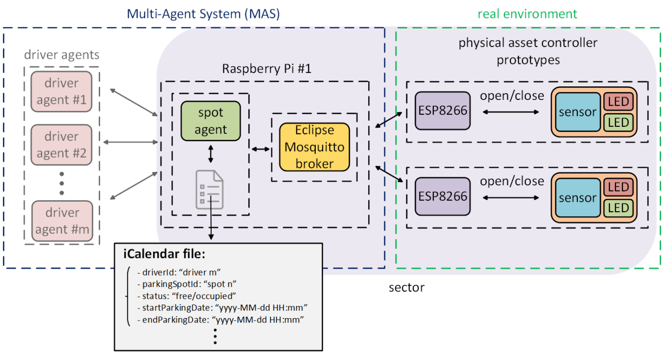

Figure 7: Implemented cyber-physical interconnection for the parking system of cars.

Figure 7 also shows that both spot agent and broker are embedded in the Raspberry Pi #1, and the broker was implemented by using the Eclipse Mosquitto. The physical prototype considers a sector with two parking spots, each one controlled by the logic control running in the ESP8266 microcontroller. The ultrasonic sensor, connected to the ESP, is placed on the ground and allows to detect the presence of a car. In fact, the sensor measures the distance between the ground and the roof, which is affected with the presence or not of a car.

Once a car arrives in the parking spot, a change in the distance measured by the sensor is detected and the following steps are taken:

- The logic control running in the ESP8266 publishes in the 'spot1/parked' broker topic that a car is parked.
- The spot agent checks the iCalendar file of that spot and updates the spot status to busy.
- The spot agent publishes in the 'spot1/message' broker topic that the reservation is confirmed.
- The logic control running in the ESP8266 receives the message and switches on the red LED.

Notice that if the reservation is not confirmed, i.e. there is no reservation in the iCalendar, the spot agent will not publish the confirmation of the reservation and therefore the logic control will not change the LED to red but keep it green, instead (it is assumed that the driver will respect the system policy and request a new parking spot to properly use the off-street parking).

When a parked car leaves the parking spot, a change in the distance measured by the ultrasonic sensor is detected by the logic control, being necessary the execution of the following steps:

- The logic control running in the ESP8266 publishes in the 'spot1/parked' broker topic that a car is no longer parked.
- The spot agent receives the message and updates the spot status to free.
- The spot agent publishes in the 'spot1/message' broker topic that the status is updated.
- The logic control running in the ESP8266 receives the message and switches on the green LED.

If the spot is free, the logic control running in the ESP8266 keeps measuring the current distance every five seconds until a change in the distance is detected. When a car arrives, the aforementioned steps are repeated.

V. Experimental Results and Discussions

The implemented agent-based CPS parking systems were operating properly during the experimental tests, being the proposed architecture successfully adapted to the different requirements imposed by parking different types of vehicles, namely bicycles and cars. In this case, the structure of the logic layer (including the MAS) is maintained, and the major changes take place exclusively in the low-level control layer (specifically in how the vehicle accesses the parking spot, e.g., using a latch, sensor or gate).

As expected due to the inherent MAS characteristics, it was noticed the robustness and scalability of the parking system, since new parking spots can be easily added or removed on-the-fly into the parking system, and without the degradation of the system performance. The reorganization of the parking system, using the holonic principles, e.g., considering different organizational structures to cluster parking spots, allows an improvement of the system efficiency.

In terms of response time, and aiming to evaluate the performance of the practice interfacing the software agent and the physical control device, the Round Trip Time (RTT) was measured along the system operation. RTT is the time measured from the moment that the spot agent publishes a message in the broker and the physical asset controller gets the notification, as illustrated in Figure 8.

Figure 9 summarizes the achieved RTT values for each communication between the spot agent and the physical asset controller during the system operation, being possible to conclude that the response time presented satisfactory and deterministic values (i.e. an average value of 51 milliseconds and a standard deviation of 0.04).

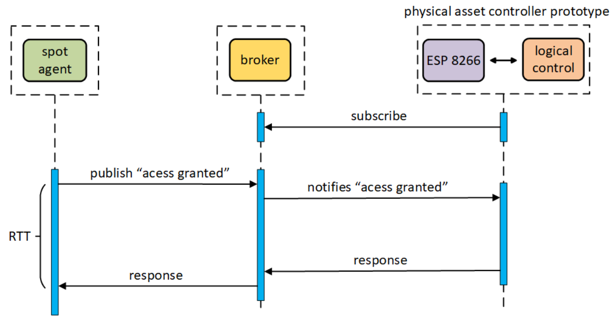

Figure 8: Process to measure the response time.

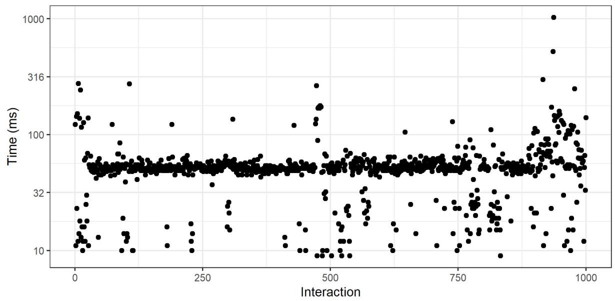

Figure 9: Response time for the spot agent to communicate with the physical asset controller.

In fact, only some few values showed deviations from the average, but this is justified by the network traffic, a factor that influences the performance of the remote interface practice approaches [8].

VI. Conclusions

A smart parking system is a suitable solution to reduce the time search for a parking spot, avoiding the traffic congestion, fuel costs and environmental pollution. For this purpose, the integration of emergent technologies, such as MAS and IoT, is fundamental. Such smart parking constitutes a CPS, allowing the interaction with the physical world through the interconnection between the software agents and the physical asset controllers to regulate the access to the parking spots.

This paper presents the development of an agent-based CPS for smart parking systems, with a special focus on the interconnection between the software agents and the physical parking asset controllers. The cyber-physical interconnection uses an interface practice based on Internet of Things (IoT) technologies and a publish-subscribe model to address the loosed-coupled requirements.

The designed agent-based CPS was implemented in two prototype parking systems, one specific for bicycles and another for cars, showing the adaptability to consider the different requirements imposed by parking different type of vehicles. Both agent-based solutions operate properly, showing important characteristics for such large-scale systems, namely in terms of modularity, scalability and reconfigurability. In fact, the parking system is build up a modular structure, where a parking spot comprises a spot agent and a physical asset controller, being possible to combine several parking spots into a composed parking spot, e.g. a parking sector, following the holonic principles. The scalability and reconfigurability are easily achieved on-the-fly by adding new spot and driver agents without the need to stop, reprogram and restart the entire system.

The response time is not the major requirement to be considered in a parking application. However, the achieved response time is quite deterministic, around 51 milliseconds, which is suitable for the parking system operation.

The future work will be devoted to extend the physical implementation of the agent-based CPS, considering a large number of parking spots, as well as to improve the reasoning capabilities embedded in driver and spot agents to support the negotiation process to reserve parking spots.

References

[1] G. Yan, W. Yang, D. B. Rawat, and S. Olariu, “Smartparking: A secure and intelligent parking system,” *Proceedings of the IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 1, pp. 18–30, 2011.

[2] Y. Geng and C. G. Cassandras, “New “smart parking” system based on resource allocation and reservations,” *Proceedings of the IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1129–1139, Sep 2013.

[3] T. Pham, M. Tsai, D. Nguyen, C. Dow, and D. Deng, “A cloud-based smart-parking system based on internet-of-things technologies,” *Proceedings of the IEEE Access*, vol. 3, pp. 1581–1591, 2015.

[4] H. Wang and W. He, “A reservation-based smart parking system,” *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS’11)*, pp. 690–695, Apr 2011.

[5] E. A. Lee, “Cyber physical systems: Design challenges,” *Proceedings of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC’08)*, pp. 363–369, May 2008.

[6] N. Jazdi, “Cyber physical systems in the context of industry 4.0,” *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR’14)*, pp. 1–4, May 2014.

[7] M. Wooldridge, *An Introduction to Multi-Agent Systems.* John Wiley and Sons, 2002.

[8] L. Sakurada, J. Barbosa, and P. Leitão, “Deployment of industrial agents in a smart parking system,” *Proceedings of the IEEE 17th International Conference on Industrial Informatics (INDIN’19)*, July 2019.

[9] R. G. Smith, “The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver,” *Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, 1980.

[10] K. Omote and A. Miyaji, “A practical english auction with one-time registration,” in *Information Security and Privacy*, ser. Lecture Notes in Computer Science, 2001, vol. 2119, pp. 221–234.

[11] T. E. Rockoff and M. Groves, “Design of an internet-based system for remote dutch auctions,” *Internet Research*, vol. 5, pp. 10–16, 1995.

[12] B. Alves, G. Alves, P. Leitão, and A. Borges, “Experimentation of negotiation protocols for consensus problems in smart parking systems,” *Proceedings of the 9th International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS’19)*, 2019.

[13] A. Koestler, *The Ghost in the Machine*. Arkana Books, London, 1969.

[14] iCalendar, “The iCalendar Standard,” 2019, last accessed 20 May 2019. [Online]. Available: https://icalendar.org/

[15] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. John Wiley Sons, 2007.

[16] Mosquitto, “Eclipse Mosquitto An open source MQTT broker,” 2019, last accessed 19 May 2019. [Online]. Available: https://mosquitto.org/