Workshops at 18th International Conference on Intelligent Environments (IE2022)
H.H. Alvarez Valera and M. Luštrek (Eds.)

doi:10.3233/AISE220046

Embedded Agents for the Development of Smart Devices on the Internet of Things

Juan A. HOLGADO-TERRIZA [a, 1], José CAICEDO-ORTIZ [b] and Pablo PICO-VALENCIA [c]
[a] *Software Engineering Department, University of Granada, Granada, Spain,*
[b] *Department of Computer Science and Electronics, University of La Costa, Barranquilla, Colombia.*
[c] *Programming and Development of Software Department, Pontifical Catholic University of Ecuador, Esmeraldas, Ecuador.*

Abstract. The Internet of things (IoT) is currently contributing to a growing range of applications for intelligent environments by means of the interconnection of heterogeneous IoT smart devices over the Internet. Generally, smart devices are implemented through services (e.g., REST webservices) which can be consumed by other services or any other external application. In any case, services are essentially passive requiring necessarily another consuming software entity to be use. In this paper, the development of a new class of smart IoT device, called smart agent-based IoT device is proposed based on the implementation of software embedded agents. These agents can be executed proactively exploring the services, resources, devices, and even other agents located in the IoT environment to enhance the functionality of the smart agent-based device. In this way, we can improve overall system performance at runtime in a unaware IoT environment where IoT devices and available resources are not known a priori. In this paper, a description of the embedded agent model and the smart agent-based IoT device are outlined as well as how it is applied in the context of a smart home environment.

Keywords. Software embedded agent, Internet of Things, smart device, smart home.

1. Introduction

The Internet of Things (IoT) describes systems composed of heterogeneous connected objects or things that collect and share their resources over the Internet. As consequence, IoT is contributing to a growing applications range for smart environments such as the remote monitoring of IoT devices by tracking some parameters of the environment, the intelligent management of the infrastructures, buildings or any other facility improving key factors such as productivity or energy efficiency, or the predictive maintenance by analysis data provided by IoT devices to predict outcomes and automate actions, among others [1].

In this context, IoT objects or devices are the first-class components responsible to perceive and collect data from the environment. Then, data are processed and stored on central servers in a cloud [2], on gateways or even directly on IoT devices in the case of

[1] Corresponding Author: Software Engineering Department, University of Granada, C/ Daniel Saucedo Aranda, s/n. E18014 Granada (Spain); Email: jholgado@ugr.es

a fog or edge computing [3], respectively. Based on the way in which the data are processed, the system can provide recommendations to users as well as make decisions by predicting users' actions. Artificial Intelligence (AI) plays an important role in carrying out these actions.

One of the AI techniques that enables active behavior to the connected devices themselves in IoT networks are software agents and multi-agent systems (MAS) [4]. There are many works in the literature that provides different approaches to integrate the agent technology into IoT ecosystems, the so-called IoT agentification process [5–8].

Several works have been presented for developing embedded agents in IoT devices. In some approaches the integration of embedded agents with resource-limited IoT devices is achieved decoupling agents from IoT devices and then creating a representative agent responsible to interact with IoT device using a M2M protocol such as MQTT. For instance, in [8], agents are executed into a MAS ecosystem on a JADE platform in a way that specific agents are responsible to manage data collected from resource-limited IoT devices through a MQTT protocol, not on the IoT devices. In a similar way, in [6], an agent-oriented core infrastructure is developed to enable flexible cooperation between heterogeneous IoT devices, smart devices, sensors and actuators. In this case, specific agents are responsible for managing IoT devices too.

The second approach consists of embedded agents executing directly in the IoT device by the support of an agent platform that facilitates the agent interactions. In this case, a powerful computing platform is desirable because the agent platform in general is high-memory demanding. In [5] the authors proposed the deployment of embedded agents directly in a Raspberry Pi B 2 as IoT device based on JADE platforms. However, in this case agent interactions are only defined to share data captured by sensors among all IoT devices in the network, rather than exploiting the proactive nature of agents performing processing tasks to provide additional services. Cruz et al [7] proposed an agent platform based on FIPA protocol for developing lightweight agents implemented in ANSI C that can be installed not only in PLCs but also on microcontrollers with memory constraints. Although the agent platform is oriented to develop cyber-physical systems, the agent model provides interesting dynamic features such as an internal and external reconfigurability, robustness to disturbances and high flexibility improving the adaptability to apply to users' needs in high- and low-level use cases.

This study describes the functional logic of a new class of smart IoT device based on the execution of proactive embedded agents deployed on these devices instead of the consumption of passive services deployed on a conventional IoT device. In our approach the embedded agents can be executed proactively exploring the services, resources, devices, and even other agents located in the IoT environment to enhance the functionality of the smart agent-based device. In this way, we can improve overall system performance at runtime in a unaware IoT environment where IoT devices and available resources are not known a priori. To facilitate the development of agents, a taxonomy of agents is also defined to automate monitoring environments in smart home scenarios. Finally, to validate the proposal it is applied to a smart home with a set of IoT devices.

This paper is structured as follows. Section 2 describes the proposed embedded agent model, the architecture of a smart IoT device based on embedded agents as well as the taxonomy of agents. Then, section 3 describes how an agent-based IoT device can be integrated into a smart home environment. The results are described in Section 4. Finally, Section 5 presents the conclusions.

2. Embedded Agents in the Context of the Internet of Things

2.1. Approach

The Internet of Agents (IoA) is an emerging paradigm on which the IoT can be viewed as an approach supported by intelligent software agents that efficiently manage the resources offered in IoT for both devices and users [9]. To realize the objective of the IoA, an agentification of the IoT must be carried out. Two approaches have been proposed in [10] to carry out this process of agentification, this is, using multiagent systems (MAS) and, complementary, by creating embedded agents in IoT devices.

In the first approach, a set of collaborating agents that make up the MAS is connected to IoT infrastructure to manage the network of IoT devices and their resources. In this case, the agent layer is completely decoupled from the IoT devices. Then, agents consume the resources of IoT devices by accessing and invoking its API (Application Programming Interface) or at least exchanging messages using a specific communication protocol [11]. The implementation of the MAS can be deployed on sophisticated servers or using the infrastructure provided by the cloud [10].

In the second approach, agents are embedded directly inside the IoT device. In a IoT environment we can distinguish two classes of IoT devices depending on the available hardware computing resources. Depending of the hardware computing resources available in IoT device, agents can be implemented natively in the IoT device or in a more powerful embedded device by using a machine-to-machine (M2M) protocol to the resource-constraint device. The integration of agent technology inside any computing platform requires enough memory to an install software platform that enables the deployment of agents [8].

”Things” are ordinarily implemented on resource-limited IoT devices to monitor some specific variables (e.g., temperature, energy, etc.) or to acts to their environment (e.g., switch lights on, off). Then, the restrictions in terms of size, computing capabilities (low power and limited memory resources), energy consumption, data storage or communication capacity can limit severely the programs executing in these devices. Consequently, the second option of running agents in a more powerful embedded device instead of the resource-constrained IoT device is the most common way to deploy agents on embedded devices [5–8,12]. In this case, the agents connect directly to resource-constrained device through a M2M or low-power communication protocol.

Unlike most contributions on agents in embedded systems, in this work we are interested in the development of agents that are natively embedded in the IoT device. The integration of agent technology inside any computing platform requires enough memory to install the software infrastructure and the agent platform that enables the deployment of agents. In this sense, lightweight agents' platforms such as osBrain [13], PADE[14], MaDKit, PANGEA, Mobile-C and other ones have been developed in order to create embedded agents that run with constrained resources.

2.2. Embedded Agents on IoT Devices

The development of embedded agents executing on a single IoT device provides a new capability to conventional IoT devices since they are essentially passive. Consequently, regardless of whether a client-server or pub-sub communication paradigm is applied to get access to IoT device and their resources, an external application is necessary to consume the IoT device by invoking its available services through its API or by

exchanging messages between both parties. In contrast, the presence of agents embedded in the IoT device promotes the proactive execution of applications that can monitor their environment by accessing their own sensors or any nearby IoT device and, subsequently, taking the most appropriate actions according to their specific goals.

Conceptually, an *embedded agent* in the context of IoA is an autonomous reactive entity that can be executed proactively linked to an IoT device in order to achieve specific goals. The logic of the agent is addressed by the definition of its behavior which include the code to solve certain tasks according to the goals to be accomplished.

As agents are collaborative by nature and they are executed at runtime, the support of an agent platform as a middleware is required to enable the execution of one or more agents in the single IoT device. Figure 1 illustrates the software infrastructure that a single IoT device must have to enable embedded agents.

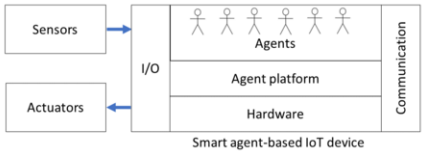

Figure 1. Embedded agents executing into a single IoT device.

The integration of embedded agents in an IoT device defines in our opinion a new generation of IoT devices smarter, more collaborative, and flexible than traditional IoT devices. As a matter of fact, an IoT device can be more adaptable to the variable conditions of the environment in which they are running and to the current presence of IoT devices available in the same context (e.g., network) performing a reconfiguration of their actions to improve the achievement of their goals. This allows to create spontaneous and reconfigurable complex agent networks from separate IoT devices that are unaware of the presence of other devices. This obviously involves seamless ways to interconnect the embedded agents hosted in different IoT devices and a lightweight middleware to deploy agents with a well-defined life cycle. This new class of IoT devices are called smart agent-based IoT (sabIoT) devices.

Conceptually, a smart agent-based IoT device is a special type of IoT device that is programmed and configured from one or more embedded agents that manage its associated resources and achievable resources on other nearby devices as well as the collaborations with other external agents whenever possible.

Another important aspect of agents is their ability to adapt their behavior based on the knowledge they can acquire through access to information sources, IoT device resources and other agents running on other IoT devices. Initially, agents may consume web services in the same way as any other application as long as it knows the IP address and port of the IoT device or dynamically by accessing a service directory where services are registered according to service-oriented architectures [12]. Moreover, agents may collaborate with other agents hosted on a different IoT device that they do not know initially using yellow pages. Figure 2 shows the basic architecture that explains how the agents from different smart agent-based IoT devices can collaborate between them. In short, the three main tasks are:

- Registering. Each agent registers its identity and its location in the network through the yellow pages when the agent is started. This allows other agents to know of its existence.

- Lookup. This action is carried out by an agent whenever it wants looking for other possible agents available in yellow pages.
- Communication. The message exchanging between two agents is only possible when both agents know the IP address and port of the agent counterpart.

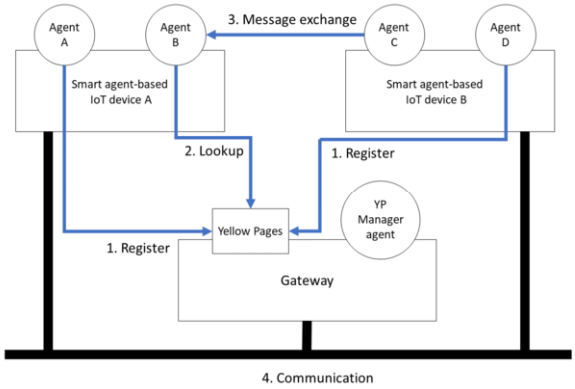

Figure 2. Basic architecture to enable the collaboration among embedded agents from different smart agent-based IoT devices over a network.

2.3. Taxonomy for Embedded Agents

In order to arrange the embedded agents running on the smart agent-based IoT device, a taxonomy of agents has been proposed in four categories: device, infrastructure, data and high-level social.

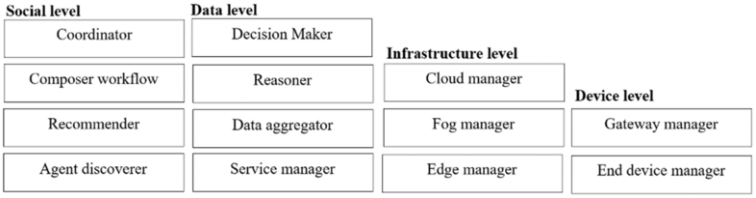

Figure 3. Taxonomy of agents for intelligent edge IoT systems.

This taxonomy is illustrated in Figure 3 showing 13 types of agents described as follows:

- End-device manager. This agent is responsible to manage directly physical resources of an IoT device.
- Gateway manager. This agent implements functionalities to manage the communications between an IoT device and other external entities connected to the network.
- Edge manager. This agent is able to process data obtained by one or more end-device manager agents by applying different algorithms or AI techniques.
- Fog manager. This agent is able to process, store temporarily and filter data that can be transmitted to another agent or kept as a temporary cache.
- Cloud manager. This agent is able to transmit data to a specific cloud through a specific connector.

- Service manager. It is an agent that can invoke any public or private service in order to execute some specific functionality or collect its resources.
- Data aggregator. This agent is able to apply pre-processing and processing techniques to generate information from raw data coming from other agents.
- Reasoner. This agent is responsible to perform logical reasoning processes of data received from other agents using formal logic mechanisms or AI algorithms.
- Decision maker. This agent is able to execute changes in the ecosystem environment where operates through the application of concrete actions.
- Discovery Agent. It is an agent able for searching counterpart agents in a directory of agents or yellow pages using syntactic or semantic searches.
- Recommender. This is an agent able to discriminate, select or recommend the most appropriate agent or agents based on the information they provide and the application of some specific criteria.
- Composer workflow. It is an agent for executing collaboratives actions using different policy types (sequentials, cooperatives, parallels) over other agents.
- Coordinator. This agent is able to coordinate agents setting cooperative actions among agents.

3. Embedded agents in a Smart Home Environment

A smart home is a typical Internet-connected intelligent environment composed of a variable number of smart devices to manage in an intelligent way the energy consumption, communications, lighting, security appliances and any other element of the home or building in order to improve the security, well-being, and comfort [15]. These smart devices are heterogeneous in nature because they can be manufactured by different companies using heterogeneous communication protocols (Wi-Fi, ZigBee, Bluetooth, Z-wave, etc.). Therefore, a residential or home gateway is commonly required for centralizing their communications, managing the interoperability, analyzing the information collected from home environment, handling the local and remote resources management and, finally, optimizing the energy consumption and bandwidth, among others [16]. Figure 4 shows a typical architecture for a smart home environment composed of smart objects interconnected through different wireless radio frequency communication protocols, such as WI-FI, Zigbee or Bluetooth.

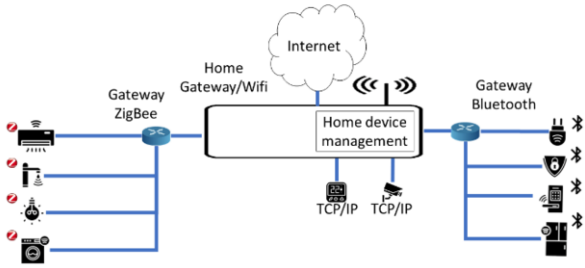

Figure 4. Conventional smart home environment with heterogeneous devices and protocols.

Integrating a sabIoT device into a smart home environment can offer many advantages over traditional home devices. In fact, the sabIoT device plays an active role in the IoT

environment thanks to the set of running embedded agents hosted on the device. These agents can proactively initiate different types of behavior related to the main purpose of the device, exploring the agents, services, and resources available in the environment. For instance, they can adapt to the conditions of that environment, improving the performance of the services or operations they provide.

Two possible scenarios can be defined when a sabIoT is integrated into a smart home environment. These scenarios are the following:

Scenario 1: *Integration of a sabIoT device into a smart home*. In this case, the sabIoT device is integrated into the smart home without modifying the smart home environment. Then, no action is necessary to do in the smart home in this case. The embedded agents on the IoT device can do the following tasks:

- Exploration and recognition of the services and resources available in the environment though the access to home gateway.
- Monitoring the environmental conditions using the sensors and actuators available in the IoT device.
- Analysis of the local and remote information collected by embedded agents.
- Execution of operations and services related with the purpose of the system adapted to the information sources achieved by agents from services or directly from the sensors/actuators.

Scenario 2: *Integration of a sabIoT device into a smart home with sabIoT devices.* The sabIoT device is integrated into the smart home where there is another sabIoT device as it is shown in Figure 5. Unlike the scenario 1, in this case, we need to carry out some reconfiguration in the IoT environment in order to facilitate the collaboration between the embedded agents from separate sabIoT devices. In this case:

- The installation of the yellow page support is required in the home gateway or any other computing platform to provide the list of available agents to other ones according to the architecture shown in Figure 2.
- The embedded agents of each sabIoT device performs the tasks described in the scenario 1 including the collaboration with counterpart agents.
- The execution of improving operations and services can be performed according to the main purposes of each sabIoT device.

4. Results

To validate the contributions provided by the development of sabIoT devices proposed in this paper into a smart home environment, we have developed a single sabIoT device for monitoring smartly the ambient conditions of the home according to scenario 1. A prototype of the device is performed on a Raspberry PI 4 running Raspbian OS with an ARM Cortex-172 processor with four cores at 1.5 GHz and 4 GB of RAM. In addition, a Grove Pi+ shield is included for connecting a temperature, a humidity, CO2, and a barometer sensors.

The sabIoT device proposed includes a set of embedded agents to determine the temperature, humidity, atmospheric pressure, and CO2 level, while a service manager agent is deployed to obtain the temperature values from the API (https://openweathermap.org/), The system looks for improving the monitoring of the ambient conditions of the room where the device is installed. It also provides specific additional services for the home inhabitants such as analyzing the outdoor environmental conditions by collecting data from the specific location of the home, The sabIoT device

verifies the status of climate device to determine the best ambient conditions, recommending to the inhabitants' different actions to perform, as open windows when the CO2 level is above a threshold, among other. The execution to these additional services depends on the possibility to find other IoT devices available in the smart home and the information collected from the environment. Table 1 shows the implemented agents and their types according to taxonomy defined in section 2.3.

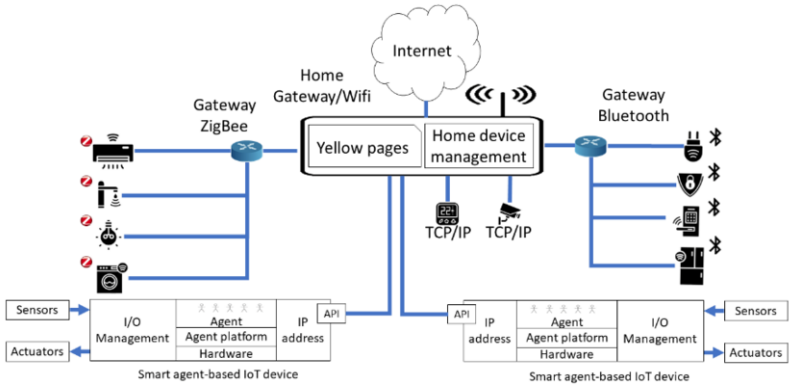

Figure 5. In this scenario the sabIoT device allows the interconnection of its embedded agents with other embedded agents of other sabIoT devices.

Once the sabIoT device is developed, we tested its validity into the laboratory where a smart home is installed with several home devices and an openhab implementation for the home gateway according to the scenario 1.

Regarding the agent technology, we can employ osBrain for the implementation of embedded agents because it is lightweight agent platform. In fact, osBrain is developed in Python and uses Pyro4 as the communication paradigm based on RPC and ØMQ; basically, it provides the support of management and message exchanges in an efficient and flexible way. This platform uses different communication patterns such as Push-Pull, Request-Reply and Publish-Subscribe, managing communications through TCP, IPC and Inproc transport protocols. The developed embedded agents are registered directly in a nameserver (the agent platform executed on sabIoT device) that acts as a local yellow pages service. This allows the communication and data exchange between agents hosted in the same sabIoT device through proxy-based connections. The communication with external agents is provided by the configuration of an external yellow pages service ina gateway in the same LAN network.

Table 1. List of agents implemented in the sabIoT device.

Agents	Purpose	Taxonomy
TempAgent	Agent for monitoring the ambient conditions of the room.	End-device manager
CO2Agent	A CO2 monitoring agent	End-device manager
AtmosPressAgent	A monitoring agent for the barometer	End-device manager
Air-condAgent	An agent monitoring the condition of the air-conditioning system.	End-device manager
ServiceTempAgent	An agent that queries the temperature via a web service	Service manager
Recommender	A recommender agent for users	Recommender
DiscoverAgent	A discover or recognition agent of other agents	Discovery Agent.
ConnectionAgent	An agent for agent-to-agent connections	Gateway manager

Preliminary results have been carried out to determine the execution of embedded agents hosted in the sabIoT device as well as the collaboration between embedded agents in the same device. In this case the embedded agent can access to internal resources (temperature, CO2, barometers sensors) and to external resources (weather service) by consuming restful services. The interconnection between embedded agents in the same sabIoT device is working too.

Some tests are carried out including another sabIoT device in the environment provoking a reconfiguration of the system. In this case, we found some network problems in the communication between embedded agents on separate sabIoT devices because a misconfiguration in the nameserver of agent platforms. We are working to simplify the process as well as improving the security in the message exchange.

5. Conclusions

This paper presents a new class of IoT smart devices based on the development of embedded agents on a lightweight agent platform that may be executed on a set of resource-constrained devices based on computing resources. The embedded agents can initiate autonomously and proactively different types of exploration, processing and prediction tasks, selecting the most appropriate resources and web services available in the case only a sabIoT device is activated in the system. When several sabIoT devices are activated, then the agent interactions between sabIoT are also enabled.

Unlike other embedded agent approaches, our proposal provides some differential features as follows: (a) Automatic exploration and recognition of conventional IoT devices and sabIoT devices with agents, physical or web service resources, which help to achieve agent goals improving that process; (b) Automatic and intelligent adaptation of embedded agents to the environment based on the information and experience with other agents improving the system execution. As a consequence, it is possible to improve the overall intelligence of IoT networks, promoting collaborative tasks between agents deployed in IoT devices, facilitating the discovering of web or local services offered by other agents and achieve a better adaptability to environmental conditions.

The embedded agent model is based on reactive type agents of different nature according to a taxonomy of 13 types of agents at four levels: device, infrastructure, data and social. Taking advantage of this taxonomy, a previous planning of the system is achieved in terms of the agents, their roles, communications and interactions among them, the information to be handled, connections established between agents and finally a correct identification of the final devices that can provide relevant information to achieve the established objectives.

To evaluate the implementation of the smart agent-based IoT (sabIoT) device, a testbed was performed using a Raspberry PI and the osBrain platform to assess its suitability in a controlled smart home environment, with several home devices and a openhab implementation for home gateway. The correct execution and interaction of several agents in the same device was validated, with the ability to acquire RESTful services and information from temperature, CO2 and barometer sensors.

Finally, the current implementation of embedded agents has some limitations relating to the communication paradigm Pyro 4 employed in osBrain agent platform. Pyro4 does not encrypt the data sent over network, so we have to work on trusted networks or on properly encrypted/safe communications [13]. Therefore, it is working on improving

information security, as well as on the solution for interaction between different sabIoT device embedded agents.

As future works, the agent platform based on osBrain will be revised and evaluated for different resource-limited devices based on microcontrollers. In addition, a complex smart home scenarios will be selected as case study to study the applicability of our proposal.

References

[1] Ullo SL, Sinha GR. Advances in Smart Environment Monitoring Systems Using IoT and Sensors. Sensors. 2020 May;20(11):3113.

[2] Firouzi F, Farahani B. Architecting IoT Cloud. Intelligent Internet of Things. 2020;173–241.

[3] Mehmood MY, Oad A, Abrar M, Munir HM, Hasan SF, Muqeet HAU, et al. Edge Computing for IoT-Enabled Smart Grid. Security and Communication Networks. 2021;2021.

[4] Savaglio C, Ganzha M, Paprzycki M, Bădică C, Ivanović M, Fortino G. Agent-based Internet of Things: State-of-the-art and research challenges. Future Generation Computer Systems. 2020 Jan;102:1038–53.

[5] Zouai M, Kazar O, Bellot GO, Haba B, Kabachi N, Krishnamurhty M. Ambiance intelligence approach using IoT and multi-agent system. International Journal of Distributed Systems and Technologies. 2019 Jan;10(1):27–55.

[6] Kato T, Chiba R, Takahashi H, Kinoshita T. Agent-oriented cooperation of IoT devices towards advanced logistics. Proceedings - International Computer Software and Applications Conference. 2015;3:223–7.

[7] Cruz Salazar LA, Mayer F, Schütz D, Vogel-Heuser B. Platform Independent Multi-Agent System for Robust Networks of Production Systems. IFAC-PapersOnLine. 2018 Jan;51(11):1261–8.

[8] de La Iglesia DH, González GV, Mendes AS, Jiménez-Bravo DM, Barriuso AL. Architecture to Embed Software Agents in Resource Constrained Internet of Things Devices. Sensors. 2018 Dec;19(1):100.

[9] Pico-Valencia P, Holgado-Terriza JA, Senso JA. Towards an Internet of Agents model based on Linked Open Data approach. Autonomous Agents and Multi-Agent Systems. 2019 Mar;33(1–2):84–131.

[10] Pico-Valencia P, Holgado-Terriza JA. Agentification of the Internet of Things: A systematic literature review. International Journal of Distributed Sensor Networks. 2018 Oct;14(10).

[11] Holgado-Terriza JA, Pico-Valencia P, Garach-Hinojosa A. A Gateway for Enabling Uniform Communication Among Inter-Platform JADE Agents. In: Intelligent Environments 2020. IOS Press; 2020. p. 82–91.

[12] Pico-Valencia PA, Holgado-Terriza JA. An Agent Middleware for Supporting Ecosystems of Heterogeneous Web Services. Procedia Computer Science. 2016;94:121–8.

[13] osBrain - 0.6.5 [Internet]. https://osbrain.readthedocs.io/en/stable/index.html (accessed Apr. 23/2022).

[14] Python Agent DEvelopment framework. Multi-agent Systems for Python Language - PADE [Internet]. https://pade.readthedocs.io/en/latest/# (accessed Apr. 23/2022).

[15] Samuel SSI. A review of connectivity challenges in IoT-smart home. 2016 3rd MEC International Conference on Big Data and Smart City. 2016 Apr;364–7.

[16] Peng Z, Kato T, Takahashi H, Kinoshita T. Intelligent home security system using agent-based IoT devices. 2015 IEEE 4th Global Conference on Consumer Electronics. 2016 Feb;313–4.