Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 238 (2024) 404–411

Procedia Computer Science

www.elsevier.com/locate/procedia

The 15th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 23-25, 2024, Hasselt, Belgium

Agents for automatic control of sensors using Multi-Agent Systems and Ontologies: A scalable IoT architecture

Herminio Paucar Curasma[a,*], Che Fan Pan[a], Julio Cezar Estrella[a]

[a]*Institute of Mathematical and Computer Sciences (ICMC-USP), Av. Trabalhador São Carlense, São Carlos, Brazil*

Abstract

Research efforts focused on Smart Building (SB) development have concentrated on the automation of resources within intelligent environments towards an enhanced experience for occupants. The process converts everyday manual activities into automatic actions, such as turning on lights upon entering a room, activating air conditioning on hot days, and switching off a television when there are no viewers. This article addresses a case study on context-aware monitoring conducted at the Laboratory of Distributed Systems and Concurrent Programming (LaSDPC[1]) of the University of São Paulo. The focus is on a Fog layer of the IoT operating closer to sensors and reducing communication delays. Concepts of context-aware systems, multi-agent systems, ontology, and MQTT protocol were considered for the implementation of the intelligent system on-site. The programming languages used were Java and Python, leveraging libraries and frameworks dedicated to such technologies. A scalable system that does not compromise computational resource use and maintains responsiveness with low data exchange latency is proposed and tests checked the intelligent behavior of the laboratory under specific conditions using temperature, luminosity, and presence of a person as parameters.

Keywords: context-aware; software architecture; smart building; multi-agent system; iot

1. Introduction

The advancement of the Internet of Things (IoT) has provided people's daily activities with higher efficiency. The proliferation of internet-connected devices such as TVs, refrigerators, and watches has facilitated data collection and development of studies and observations, thus, optimizing tasks. Examples [14][15] such as remote monitoring and analysis of patient health conditions[10] and, in the domestic environment, the energy consumption of electronic devices can be monitored and controlled.

[1] http://lasdpc.icmc.usp.br/

* Corresponding author. Tel.: +55-16-935001794.
E-mail address: herminiopaucar@usp.br

10.1016/j.procs.2024.06.041

The growing expansion of such technology has led to the implementation of smart city projects[1], i.e., cities whose structures and buildings are connected to networks, generating and collecting data for improving their inhabitants´ life quality[4]. A smart city consists of Smart building constructions that incorporate Fourth Industrial Revolution technologies, including Cloud Computing and IoT for primarily optimizing and automating resource usage, especially in environments with high human traffic [12][5].

Places regularly frequented require resources for maintaining the comfort and convenience of visitors [13]. As an example, let us consider a university computer laboratory, used by students for disciplines that require practical applications towards enhancing learning. Such use implies high energy costs for the maintenance of devices such as computers, air conditioners, lights, televisions, and operational equipment [3]. The implementation of smart building concepts makes those resources' efficient and automatic use viable through the integration of sensors to monitor environmental variables, as claimed by [6].

This study models and analyzes a scalable context-aware Multi-Agent System, where data obtained by sensors enable automatic decision-making for device changes, such as turning on/off air conditioners and lights. The system will be applied within the context of a Smart Building represented by LaSDPC. In what follows are the main contributions of the study:

- Proposal of a basic software architecture for the autonomous control of sensors in a Smart-Building context.
- Presentation of the functionalities that a scalable architecture should have.
- Evaluation of the system's scalability considering number of sensors and characteristics of the platform where the system will be applied (memory, CPU, among others).

It must be highlight that this project is part of large project that includes Reactive Manifesto principles, Reactive Microservices, Digital Twin, and Federated Learning (see this site[2] for the complete project).

2. Related Work

The studies were gathered from the most renowned technology research libraries such as IEEE, Scopus, WoS, and ACM and an specific search string *('smart building' AND ontology AND 'multi-agent system')* was employed, resulting in an extensive selection of studies. Articles proposing software architectures and those whose architectures had been tested in real or simulated scenarios were prioritized.

[7] implemented a real-time quantification mechanism for the shelf life of bananas. According to the data, up to 60% of bananas are discarded from production to the final consumer. Therefore, the study proposes applying IoT knowledge through Arduino in conjunction with temperature, humidity, and luminosity sensors. Neural networks predicted the moment of banana spoilage and multi-agent systems detected devices (Arduino) automatically, facilitating communication between applications and those devices.

[9] employed an MAS and an ontology to enhance MobiLEHealth (Mobile Learning Environment for Health) system, which provides individuals with chronic diseases with a deeper understanding of their health conditions, improving their quality of life. The primary goal of such enhancement was to adapt the system's graphical interface to meeting the needs of the target audience, predominantly composed of elderly diabetic individuals, who demand an easily understandable and navigable interface due to visual, cognitive, and motor limitations resulting from age and illness.

In a more specialized context of ontology, [6] served as the basis for the definition of rules for guiding the intelligent behavior of agents in LaSDPC project. The article describes the conception of a context-aware intelligent system employing ontology to optimize energy consumption in buildings. The authors developed Onto-SB, a widely comprehensive ontology focused on smart buildings and encompassing concepts related to buildings, individuals, user profiles, devices, and energy.

Several inference rules were established, including activity detection such as watching television, calculating device energy consumption, deactivating lights in unoccupied rooms, and turning off the heater when the temperature

[2] https://smart-lasdpc.github.io/about-project.html

exceeds 25°C. A simulator called Open-SBS was developed to evaluate the system, reproducing a real scenario in an environment simulating a family-inhabited residence. The results were satisfactory, demonstrating the system reduced approximately 40% of the total energy consumption.

Table 1 shows a comparison of the three studies related to the present proposal.

Table 1. Comparison of methods and approaches among related studies.

	Study 01 [7]	**Study 02** [9]	**Study 03** [6]	**Our proposal**
MAS	Yes	Yes	No	Yes
Framework	JADE	JADE	-	JADE
Context-aware	Yes	Yes	Yes	Yes
Model used	Neural networks	Ontology	Ontology	Ontology
Inference rules	-	Yes	Yes	Yes
Test conducted	Practical	Practical	Simulated	Simulated

The studies proved fundamentally important for expanding the understanding of the application of multi-agent systems for context-aware monitoring and development of ontologies and inference rules. Although they addressed different contexts, they enabled extraction and adaptation of the concepts employed to fit the reality of LaSDPC. Therefore, they played an essential role in guiding the system's development within the laboratory scope.

3. Method

The multi-agent system developed comprises four types of agents, namely, *Creator*, *Monitor*, *Collector*, and *Rational*, of which each plays a specific role and their cooperation enables the creation of an intelligent system for the laboratory. *Creator* receives sensor data and creates *Monitors* for overseeing a group of sensors. *Collector* gathers data from each sensor and, upon finishing collection, sends them to *Rational*, which uses them to manipulate the ontology.

The architecture is displayed in Figure 1, with a focus on the Fog layer, where sensors, controllers, and smart devices send their data to a Smart Gateway (represented by Eclipse Kura[3]) containing MQTT broker through a WiFi network inside LaSDPC. From this point, *Creator* and *Collector* connect to the broker towards receiving the data.

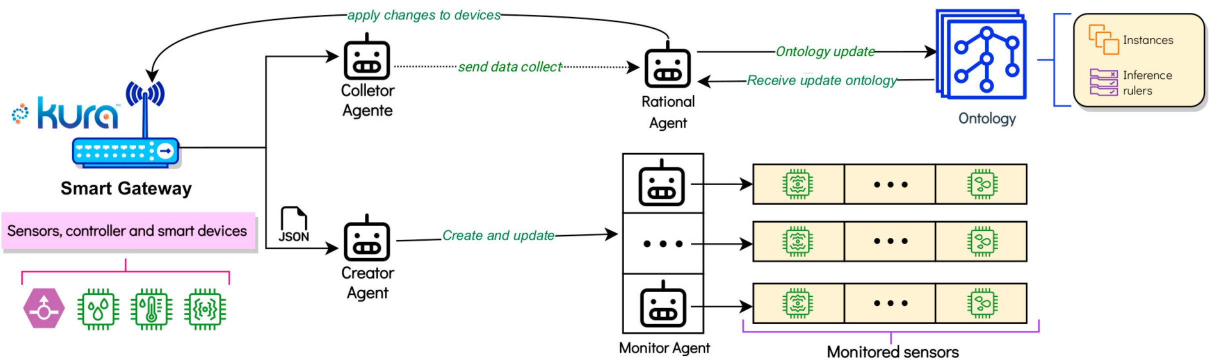

Fig. 1. Multi-agent system architecture.

Except *Rational*, which is created in Python with the use of Owlread2 library, each agent is developed within JADE framework. The choice for Python is attributed to its handling of ontologies. Whereas OWL API in Java is used for the same purpose, Owlread2 provides a more direct and simplified approach for the creation of instances and update of values in the ontology. Additionally, it offers comprehensive, detailed, and up-to-date documentation, covering all

[3] https://eclipse.dev/kura/

available resources and facilitating the implementation of *Rational* agent functions. Therefore, Owlread2 was chosen due to its advantages. In what follows are details on how Owlread2 works.

Initially, sensors and objects transmit their data to the broker within Smart Gateway. The information is formatted in JSON that contains id (Unique identifier of the sensor), type (Sensor type, e.g., temperature, light, or motion), date (Date and time of the sensor reading), agent_id (*Monitoring Agent* responsible for overseeing the sensor), measure (Type of value read by the sensor), and state (Current state of the sensor i.e., ON or OFF). The sensor simulator scripts can be seen on the project repository[4].

Collector and *Creator* receive the data from the broker and *Collector* stores the information until it gathers data from all types of sensors present in the laboratory.

Once the collection has been completed, the data are sent to *Rational*, which manipulates the LaSDPC ontology and, according to the data, creates instances for objects and sensors, representing the states and values at the time of reading. Subsequently, inference rules are applied.

If a rule is triggered, the ontology is updated, sending instructions for the update of the state values in each sensor's controller (physical devises), demonstrating the system's context-awareness, for decisions are made autonomously with no human intervention. The next rule is then tested and the process continues until all rules have been processed, after which the data cycle restarts with the collection of a new round of data by *Collector*.

Creator receives the data and checks if any sensor lacks a *Monitor* agent for supervision. *Monitors* have a maximum number of sensors they can oversee, which is determined by the laboratory administrator. If a sensor has no assigned agent for monitoring, *Creator* queries the system for an available *Monitoring* agent. In case of no availability, *Creator* generates a new *Monitor* within the system and assigns the sensor to it. Alternatively, if space for monitoring is available, the sensor is allocated to an agent with capacity, demonstrating the system's capability for automatic sensor management, operating seamlessly with no human intervention.

Table 2 shows the behaviors of the agents in the system, which has proven autonomous and intelligent when managing the sensors.

Table 2. Behavior of agents

Automatic sensor management	**Ontology manipulation**
Connection with the broker, via MQTT protocol by *Creator*; Automatic detection of sensors by *Creator*; Receipt of sensor data by *Creator*; Creation of a new *Monitoring* by *Creator*, whenever necessary; When a *Monitor* does not have sensors to monitor, it is terminated; Each *Monitor* has the objective of operating with the maximum number of sensors established or getting as close to this number as possible; If a *Monitor* is terminated with sensors, it is recreated with the same ID and receives the sensors again.	*Collector* initiates together with *Creator*; *Collector* collects data from each sensor until it obtains data from all; *Collector* transfers the collected data to *Rational*; *Rational* receives the data and populates the ontology with instances; *Rational* applies the inference rules to the ontology; *Rational* updates the ontology.

Graphical interface for the visualization of agents and their sensors:

A graphical interface was developed for a better understanding of the agents within the system and the sensors each one is responsible for, as illustrated in Figure 2. Although JADE provides a native graphical user interface for the visualization of the agents´ behavior, this GUI does not show the sensors interacting with the system's agents. Therefore, a custom interface that displays information on the agents and their respective monitored sensors was created. It shows count of present agents, number of sensors each agent has, and names of the sensors.

Ontology that represents LaSDPC: The ontology within the LaSDPC domain was constructed by Protégé software and according to the recommendations of [2]. The classes pertain to the sensors in the laboratory, such as temperature, luminosity, and motion, as well as the objects controlled through established rules, such as air conditioning, doors, blinds, and lamps. Those classes form the foundation of the ontology, enabling control over the primary functions that enhance the experience of occupants within the environment and promote energy efficiency. Figure 3 depicts the ontology.

[4] https://github.com/Smart-LaSDPC/MAS-and-Ontology-to-Smart-Building/tree/main/sensor-simulators

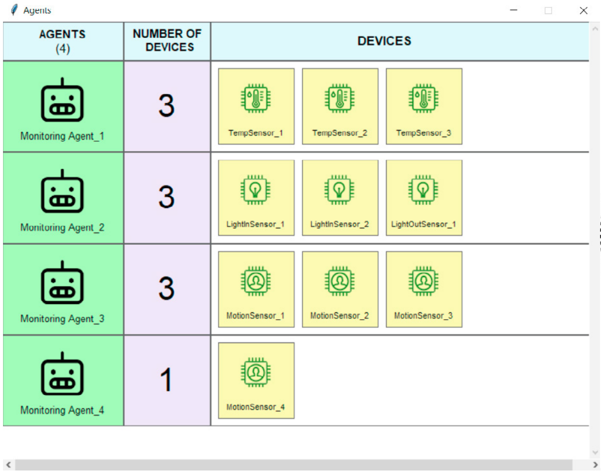

Fig. 2. GI showing the agents and their respective sensors.

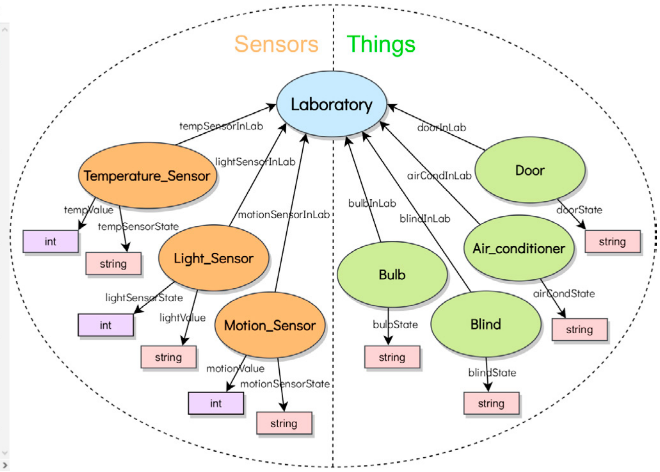

Fig. 3. Ontology developed for LaSDPC.

Regarding properties of objects within the ontology, a relationship indicating the location of each sensor and object was established. Since LaSDPC comprises two laboratories (1006 and 1008), understanding the location of each device is crucial for a better comprehension of laboratory data and mapping of the distribution of devices. Additionally, each object has a state property. For doors and blinds, OPEN/CLOSED indicate the state of being open or closed, whereas lamps and air conditioners can take ON/OFF values to represent their operational status.

Two data properties were adopted for the sensors - one reflects the sensor's state, which can be ON/OFF, whereas the other represents the values read by it. The temperature sensor records the temperature in degrees Celsius, the luminosity sensor measures the amount of lux in the area, and the presence sensor indicates presence or absence of people with values of 1 for presence and 0 for absence.

Inference Rules: The initial inference rules developed for LaSDPC are presented in more detail on the project repository[5]. Our group has been working on the development of a robust ontology that enables the execution of rules aimed at automating the state changes of objects, such as air conditioning, lamps, doors, and blinds, thus eliminating the need for human intervention. The recommendations and guidance provided by [11][6] have been followed. The functionalities aim to create a more comfortable environment for laboratory occupants, optimize resource usage, and conserve energy.

4. Results

This section addresses the tests conducted for verifying the **scalability** of the developed system. Parameters such as CPU usage, memory consumption, and message reception time by *Creator Agent* were analyzed following the recommendations of [16, 8]. The data were sent through sensor simulators, generating random values, and the inference rules of *Rational agent* were tested and checked for application when the conditions for a rule have been met.

Simulation of sensors and object states: Temperature, luminosity, and motion sensors used in the experiment were simulated in Python programming language. The simulators generate random values for environmental readings. The temperature sensor produces values between 10°C and 40°C and luminosity ranges from 0 to 500 lux. The motion sensor randomly sets value 1 for presence of people and 0 for absence. Air conditioning, lamp, door, and blind states were also simulated. The tests' initial states were air conditioning off, light on, door open, and blinds closed.

[5] https://github.com/Smart-LaSDPC/MAS-and-Ontology-to-Smart-Building/tree/main

System scalability test considering number of agents and sensors: Towards assessments of the system's scalability in terms of responsiveness, tests were conducted to evaluate the response time in data reception by the *Creator Agent*. The tests measured the time required for the receiving of data from a random sensor within the system. In larger environments, there might be a higher number of sensors and objects sending data to the agent. Therefore, the aim was to assess whether the agent could manage all information without significantly impacting the system with delays. The tests involved the use of up to 100 sensors simultaneously, sending readings every second until reaching 1000 readings sent by each sensor. The data were sent in JSON format and averaged approximately 122 characters each. Towards the analysis of a scenario that imposes higher stress on the system, the quantity of 1 sensor per Monitoring Agent was established, maximizing the number of agents in the system. The result is provided in Figure 4.

A virtual machine hosted on one of the nodes within the Andromeda cluster of LaSDPC was used for the tests with the following configurations: QEMU Standard PC model (Q35 + ICH9, 2009), AMD Opteron 63xx class CPU x4, 8GB of memory, Ubuntu 22.04.3 LTS 64-bit OS, Java SE 8, JADE 4.5.3, and Mosquitto 2.0.15.

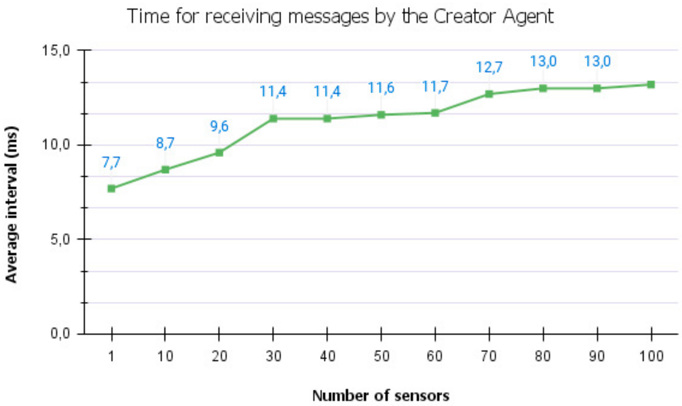

Fig. 4. System scalability analysis, focusing on measurements of message intervals.

The graph shows a tendency toward a linear growth in the average interval for the receiving of new data from a same sensor as the number of sensors increases. Therefore, with only one sensor in the system, the interval averaged 7.7 milliseconds. The growth becomes linear up to 30 sensors, where the time remains nearly constant between 40 and 60 sensors. Subsequently, there is a new increase after reaching 70 sensors, with an interval of 13.2 milliseconds with 100 sensors.

The interval for receiving sensor data did not significantly increase as the number of sensors grew. This is partly attributed to the fact the sensor's message-sending interval occurs every second, which is a value the system can handle without difficulties, following the machine's settings on which the agent operates.

Additionally, both CPU and memory usage in the machine were analyzed regarding hardware resource use. Figures 5 and 6 display the corresponding graphs for those metrics. The same number of sensors of the time test was considered and ranged from 1 to 100.

The analysis of the CPU utilization graph revealed the values did not increase suddenly, reaching peak consumption when 100 sensors simultaneously transmitted data. The results regard the consumption of 1 CPU core, considering that the utilized machine possesses four cores. Therefore, the Multi-Agent System (MAS) can be hosted on a virtual machine without overburdening it. The memory usage showed minimal fluctuation, consistently hovering around 1 GB for varying quantities of tested sensors. Although not excessively low, this value does not yet pose a constraint for the machine, equipped with 8 GB of memory. Consequently, MAS remains scalable, considering the current resources available on the machine.

Testing of Inference rules in Ontology: Temperature, luminosity, and motion sensor simulators were employed for tests with *Collector* and *Rational Agent* and are responsible for transmitting captured environmental data to the broker (eclipse Kura). Additional simulators were developed to send the states of the objects addressed in this study, namely

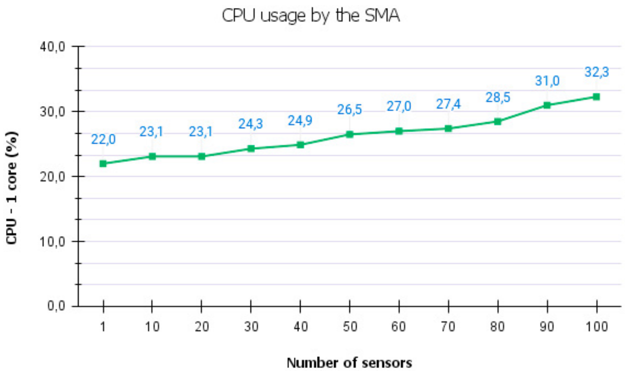

Fig. 5. System scalability analysis measuring CPU.

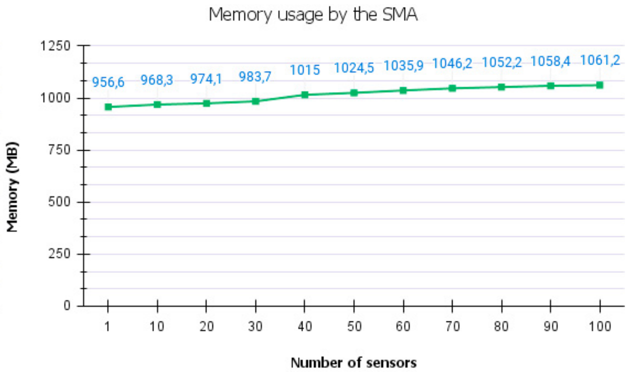

Fig. 6. System scalability analysis measuring memory.

air conditioners, lamps, doors, and blinds. Following the program execution and simulator operation, the output result, considering four rounds of data collected by *Collector* agent, is presented on the project repository[6]. It showcases the rounds of data collected by *Collector*, encompassing both sensors and objects. The data were formatted in JSON, as previously defined, and contains information about the environmental variable readings by the sensors and the state of the things.

In the *first round*, the air conditioning was off, the lamp was on, the door was open, and the blinds were closed. The sensors indicated absence of people, with 320 lux indoors, 423 lux outdoors, and 29°C temperature. Therefore, following the inference rules, *Rational Agent* applied only Rule 1, turning off the lamp due to the absence of people in the laboratory and keeping the light on, thus saving energy.

The *second round* incorporated updates in the ontology from the previous round, now with the lamp off. The remaining states indicated the air conditioning was off, the door was open, and the blinds were closed. The sensors detected presence of at least one person, with 365 lux indoors, 402 lux outdoors, and 37°C temperature. Therefore, Rule 2 was triggered due to the presence of a person(or people) and temperature above 28°C temperature, resulting in the activation of the air conditioning. Subsequently, the ontology was updated, triggering Rule 3 for the closing of the door and prevention of cold air from escaping the laboratory.

In the *third round*, the updated states of the objects were displayed, with the blinds remaining closed. Sensor data indicated presence of someone in the environment, 121 lux indoors, 348 lux outdoors, and 15°C temperature. Due to low internal luminosity and higher external luminosity, the blinds were opened to save energy and prevent the lights from being activated.

In the *fourth round*, the updated states of all objects were presented. No presence of individuals was recorded; measurements indicated 250 lux indoors, 310 lux outdoors, and 17°C temperature. Given this information and upon analyses of the defined inference rules, none of them applies to this scenario. Therefore, no object states were altered.

5. Conclusions and future work

This paper reported on the development of a system that manages the sensors of LaSDPC and uses the received data to create an intelligent environment. Concepts such as context-aware monitoring, multi-agent systems, ontology, and MQTT protocol were incorporated towards the achievement of the goal. Research on those concepts was reviewed during the development of the system, enabling their adaptation to the laboratory context. The system proved promising through the use of sensor and object simulators and can serve as a foundation for incorporating and enhancing behaviors and rules according to the needs and preferences of laboratory users. It is scalable, as evidenced in tests conducted with up to 100 sensors sending data simultaneously. This number of sensors was established for the examination of a scenario beyond the laboratory environment, since smaller spaces do not demand such a significant volume of sensors. This scalability enables an effective management of devices that measure critical environmental

[6] https://github.com/Smart-LaSDPC/MAS-and-Ontology-to-Smart-Building/blob/main/Inference_Rules_Test.md

variables such as temperature, luminosity, and presence of individuals, hence, installation of other types of sensors (e.g., sound, water leakage, and air quality). The study is expected to serve as a practical model for transforming LaS-DPC into a Smart Building with greater comfort to laboratory users. Moreover, a reduction in energy consumption is anticipated, since the system enables an efficient use of environmental resources available. Future studies aim at the integration of all project components, namely, Reactive architecture on the cloud, Federated Learning, and Digital Twin for the creation of a complete solution on Edge, Fog, and cloud computing layers. Such a unified approach will offer a comprehensive and adaptable solution applicable to all university laboratories, thereby opening new research gaps (e.g., anomaly detection, pattern discovery with data mining, among others) for future researchers to explore areas of Distributed systems and AI.

Acknowledgements

This research was developed using the computational infrastructure of the Distributed Computing Lab of ICMC-USP—the University of São Paulo (http://infra.lasdpc.icmc.usp.br/) and also with resources from the Center for Mathematical Sciences Applied to Industry (CeMEAI http://www.cemeai.icmc.usp.br/) funded by the São Paulo Research Foundation FAPESP (Project Number 2013/07375-0) and also by Project Number 11/09524-7).

References

[1] Al Dakheel, J., Del Pero, C., Aste, N., Leonforte, F., 2020. Smart buildings features and key performance indicators: A review. Sustainable Cities and Society 61, 102328. doi:10.1016/j.scs.2020.102328.

[2] Allegre, W., Burger, T., Antoine, J.Y., Berruet, P., Departe, J.P., 2013. A non-intrusive context-aware system for ambient assisted living in smart home. Health and Technology 3, 129–138. doi:10.1007/s12553-013-0050-y.

[3] Berardi, U., 2017. A cross-country comparison of the building energy consumptions and their trends. Resources, Conservation and Recycling 123, 230–241. doi:10.1016/j.resconrec.2016.03.014.

[4] Buckman, A.H., Mayfield, M., B.M. Beck, S., 2014. What is a smart building? Smart and Sustainable Built Environment 3, 92–109. doi:10.1108/SASBE-01-2014-0003.

[5] Coelho, D.F.B., Cruz, V.H.M., 2017. Edifícios inteligentes - uma visão das tecnologias aplicadas. URL: https://api.semanticscholar.org/CorpusID:185889016.

[6] Degha, H.E., Laallam, F.Z., Said, B., 2019. Intelligent context-awareness system for energy efficiency in smart building based on ontology. Sustainable Computing: Informatics and Systems 21, 212–233. doi:10.1016/j.suscom.2019.01.013.

[7] Do Nascimento, N.M., de Lucena, C.J.P., 2017. FIoT: An agent-based framework for self-adaptive and self-organizing applications based on the Internet of Things. Information Sciences 378, 161–176. doi:10.1016/j.ins.2016.10.031.

[8] Dähling, S., Razik, L., Monti, A., 2021. Enabling scalable and fault-tolerant multi-agent systems by utilizing cloud-native computing. Autonomous Agents and Multi-Agent Systems 35. doi:10.1007/s10458-020-09489-0.

[9] Freitas, A.D., Neto, F., Silva, S., Neto, A., Domingues, A., dos Santos, I., dos Santos, W., Passos, D., 2019. Implementation of an adaptive interface for a support system for diabetes patients, in: Proceedings of the XIX Brazilian Symposium on Computing Applied to Health, SBC, Porto Alegre, RS, Brazil. pp. 312–317. doi:10.5753/sbcas.2019.6268.

[10] Griffor, R, E., Greer, Chris, Wollman, A, D., Burns, J, M., 2017. Framework for cyber-physical systems: Volume 1, overview doi:10.6028/NIST.SP.1500-201.

[11] Janowicz, K., Haller, A., Cox, S.J., Le Phuoc, D., Lefrançois, M., 2019. Sosa: A lightweight ontology for sensors, observations, samples, and actuators. Journal of Web Semantics 56, 1–10. doi:10.1016/j.websem.2018.06.003.

[12] King, J., Perry, C., 2017. Smart buildings: Using smart technology to save energy in existing buildings. American Council for an Energy-Efficient Economy .

[13] Klepeis, N.E., Nelson, W.C., OTT, W.R., Robinson, J.P., Tsang, A.M., Switzer, P., Behar, J.V., Hern, S.C., Engelmann, W.H., 2001. The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants. Journal of Exposure Science and Environmental Epidemiology 11, 231–252. doi:10.1038/sj.jea.7500165.

[14] Kotha, H.D., Gupta, V.M., 2018. Iot application, a survey. International Journal of Engineering & Technology doi:10.14419/ijet.v7i2.7.11089.

[15] Minoli, D., Dressendofer, J.A., 2022. Traditional Sensor Networks/IoT Services. pp. 87–138. doi:10.1002/9781119716075.ch4.

[16] Yang, Y., Ren, F., Zhang, M., 2021. An agent-based adaptive mechanism for efficient job scheduling in open and large-scale environments. Journal of Systems Science and Systems Engineering 30, 400 – 416. doi:10.1007/s11518-021-5494-4.