

# Using Jason framework to develop a multi-agent system to manage users and spaces in an Adaptive Environment System

Pedro Filipe Oliveira<sup>1,2</sup>, Paulo Novais<sup>1</sup>, and Paulo Matos<sup>2</sup>

<sup>1</sup> Algoritmi Centre/University of Minho, Department of Informatics, Braga, Portugal

<sup>2</sup> Institute Polytechnic of Bragança, Dep. of Informatics and Communications, Bragança, Portugal

**Abstract.** Manage user preferences and local specifications on an IoT adaptive system is a actual problem. This paper uses Jason framework to develop a multi agent system to achieve a Smart Environment System, and supports interaction between persons and physical spaces, that users want to smartly adapt to their preferences in a transparent way. This work proposes a new approach, that has been developed using a multi agent system architecture with different layers to achieve a solution that entails all the proposed objectives.

**Keywords:** adaptive-system, AmI, multi-agent, IoT, Jason, Argo

## 1 Introduction

The Artificial Intelligence field continues with an exponential growth rate, especially in the different sectors applicability. Currently, multi-agent systems have been used to solve diverse situations, like in Ambient Intelligence.

Ambient Intelligence (AmI), is an ubiquitous, electronic and intelligent environment, recognized by the interconnection of different technologies/systems, in order to carry out the different daily tasks in a transparent and autonomous way for the user [6].

Thus, multi-agent systems are made up of autonomous agents present in the environment and who have the ability to make decisions derived from the interpreted stimuli as well as the connection with other agents, to achieve common goals [15].

Currently there are different languages as well as platforms for the development of this type of systems, namely *3APL*, *Jack*, *Jade/Jadex*, *Jason*, among others [4]. *3APL*, *Jadex* and *Jason* use agents with cognitive reasoning models as an alternative to more traditional reactive models.

Focusing on the cognitive model Belief-Desire-Intention (BDI), which allows the creation of intelligent agents capable of making decisions based on beliefs and perceptions, desires and intentions that the agent may have at a given moment.

Jason is a framework for the development of SMA, which has an interpreter for the *AgentSpeak* language developed in *Java*, which implements the previously

mentioned BDI model. There is also an extension for *Jason*, called *ARGO*, this being a customized architecture that allows the programming of cognitive agents using controllers (taking advantage of sensors and actuators).

There are already different works in the literature that present solutions for integrating SMA with AmI, and specifically with Smart Homes, using *Jade* [7][2], which is reactive, and using *Jason* with *JaCaMo* [8][9][1].

Projects that use *Jason* as development language are mainly simulated and there are no works in the literature on physical integration with real environments or hardware to meet ubiquitous computing that use *Jason* as SMA in the literature.

*ARGO* is an architecture that aims to facilitate ubiquitous SMA programming using *Jason*, regardless of the chosen field.

This work aims to propose an autonomous Smart Home model controlled by cognitive agents using *Jason* and *ARGO* and to manage physical devices, since *ARGO* agents allow communication with different controllers (Arduino, Raspberry).

For this, the work has a prototype of a house with six divisions, each with lighting, and a heating system.

To evaluate the development of the SMA and the prototype, a series of performance tests were performed taking into account parameters such as the number of agents, number of controllers, speed of reasoning of the agents, moment of perception of the environment and information filtering, in order to explore different implementation strategies about the system.

The main expected contribution of this work is the possibility of applying SMA to ubiquitous prototypes using the *Jason* framework and *ARGO* architecture applied to intelligent environments.

This project proposes as well, a solution using a multi-agent system. Next, the problem will be detailed, as well as a solution proposal, which includes all the architecture developed, that later will be implemented and tested.

## 2 Materials and Methods

Figure 1, shows the scenario of an environment where it intends to develop this work. Explaining this figure, it can be seen the user who through its different devices (smartphone, wearable, and other compatible) communicates with the system, and for that can be used different technologies, like Near Field Communication (NFC) [14], Bluetooth Low Energy (BLE) [3] and Wi-Fi Direct [5]. Next, the system performs communication with the Cloud, to validate the information. And then the system will perform the management of the different components in the environment (climatization systems, security systems, other smart systems).

To optimize the predictions of the solution proposed, an architecture for a multi-agent system was defined. The roles that each agent should represent, as well as the negotiation process to be taken, and the different scenarios in

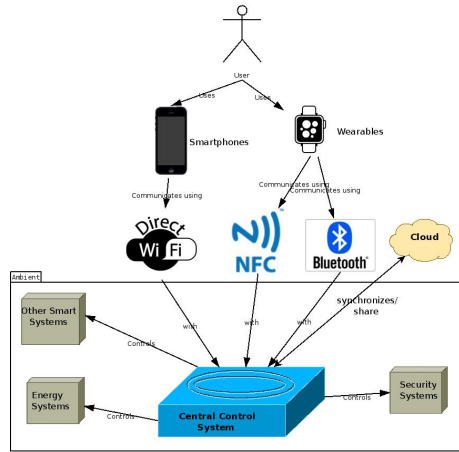


Fig. 1: Problem Statement

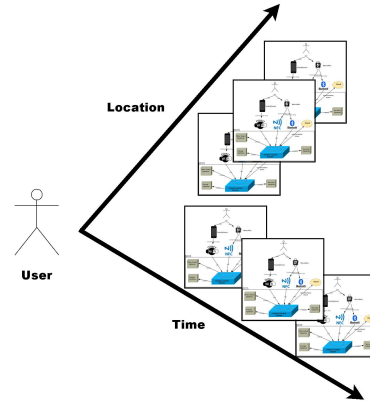


Fig. 2: Contextualization of Time/Environment Dimensions

which this negotiation should take place and the way it should be processed were specified.

For the project development, two phases are defined as follows:

- Hardware (local systems) installation;
- Multi-agent system development;

Firstly, the entire physical structure must be prepared, where the local devices (Raspberry) equipped with the network technologies previously identified, so that they can detect the users present in the space.

The comfort preferences of each user present in the space is sent to the agent every time an *ARGO* agent performs his reasoning cycle (by calling the *getPercepts* method, which must exist in all controllers that need to send perceptions to agents).

Thus, the SMA must be programmed independently from the hardware, taking into account only the actions that must be performed to achieve the ideal comfort values for the space in question, and then these values are sent to the actuators. The connection to the actuators was not taken into account in this work, implying that it is automatic and without any constraint for the user.

A prototype was thus implemented in a house, taking into account all the architecture of the SMA and the comfort actuators present in it.

For this purpose, a *Raspberry* is used per division, in this case three on the ground floor (living room/kitchen, office, bedroom) and three on the first floor (one per room).

Regarding the actuators, these divisions have a hydraulic radiant floor heating system heated by a heat pump, and a home automation system that controls the luminosity intensity in the different rooms.

Figure 3 and 4 illustrates where the different local systems (*Raspberry's*) are placed in each of the floors.

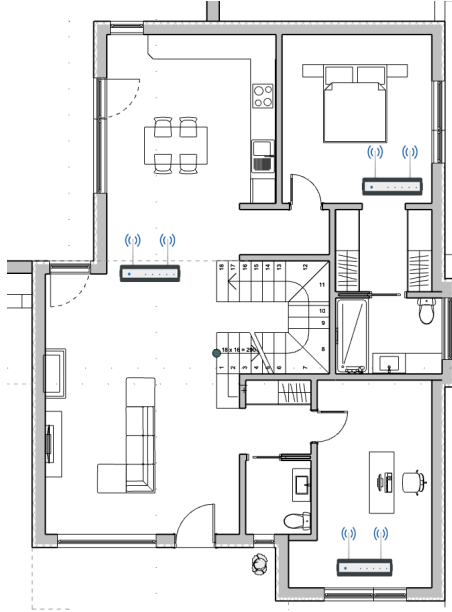


Fig. 3: Ground Floor

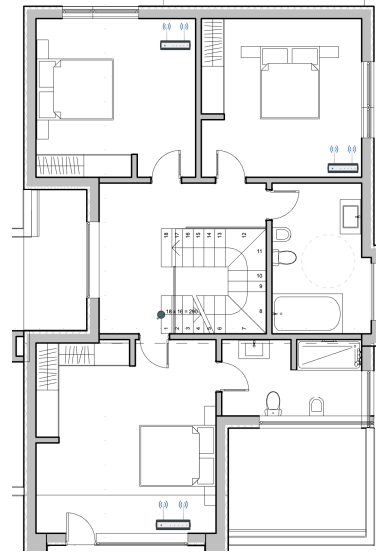


Fig. 4: First Floor

Viewing in more detail, a 3D model was designed, where the operation of the system for a specific space can be visualized, like can be seen in figure 5. Explaining the model, we can see different people present in the space, as well as the local system present in it, the arrows illustrate the autonomous communication process between the users' peripherals (smartphones) and the local system and the one with the central server (Cloud), which will allow to have the information for each of the agents to work and to reach the values of comfort preferences to use in the actuators.

This work proposes an autonomous Smart Home model, controlled through cognitive agents, which get the final information to be applied by the actuators.

For do that, a house with six divisions was prototyped with different comfort features, namely temperature, luminosity, audio and video. The considered parameters for performance evaluation are as follows:

- Number of agents used;
- Agent speed reasoning;
- Information filtering;
- Environment perception time;

This work resulted in the complete specification of an architecture that supports the solution found, to solve the presented problem. It will now be implemented, tested and validated using real case studies, so as to gather statistical information to assess its effectiveness and performance in the context of application.

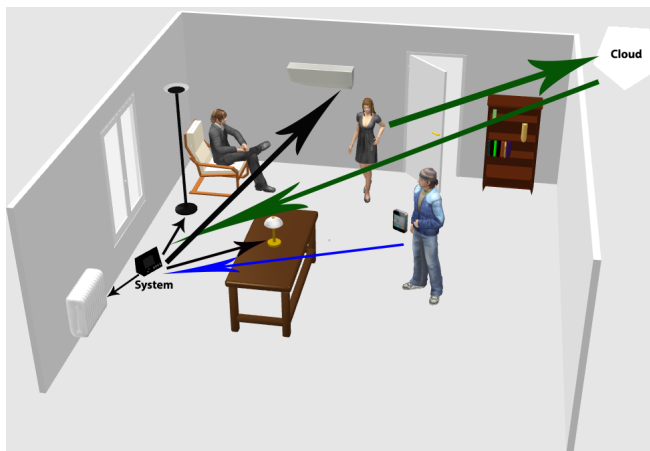


Fig. 5: Example of the System in a division

This work aims to give continuity and finalize the doctoral work presented in previous editions [10][11][12]. Thus figure 1 and figure 2 exemplifies in a global way the architecture of the system where this work has been carried out.

### 3 Results

This section presents the technologies used in this project for the development of the entire SMA applied to AmI.

Jason is a framework with its own language for the development of cognitive SMA, and using the customized architecture of ARGO agents it is possible to bridge the gap between SMA and actuators and sensors present in the real world.

Figure 6 exemplifies the use case diagram, explaining this diagram, we can verify the functioning of the different agents. Namely the information received by them, and how the negotiation process is carried out, those involved in it, and how the final result of the negotiation is passed to the actuator.

Initially, we can verify that the agent that represents the local system receives its information, namely the security information (maximum values of temperature, gases, and others). For each user present at the local, there will be an agent who represents him, he will receive information about the user preferences from the central system.

The negotiation process will then be made up of the local system agent and each of the users agents present at the local. The negotiation result of will then be passed on to the different actuators present in the local.

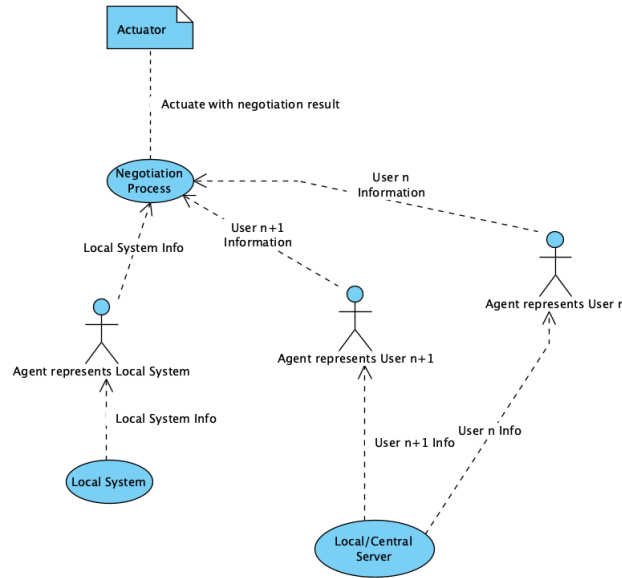


Fig. 6: AmI System - Use Case diagram

### 3.1 Framework Jason

Jason is a framework with an agent-oriented programming language, has an *AgentSpeak* Java interpreter for the development of intelligent cognitive agents using BDI.

The BDI consists of three basic constructions: beliefs, desires and intentions. Beliefs are information taken as truths for the agent, which can be internal, acquired through the relationship with other agents or through the perceptions observed in the environment. Desires represent an agent's motivation to achieve a certain objective and intentions are the actions that the agent has committed to perform.

AgentSpeak is a programming language focused on the agent approach, which is based on principles of the BDI architecture.

In addition to these concepts, the Procedural Reasoning System allows the agent to build a real-time reasoning system for performing complex tasks. Jason's agents have a reasoning cycle based on events that are generated from capturing perceptions of the environment, messages exchanged with other agents and through their own conclusions based on their reasoning.

These events can be triggered by triggers that lead to the execution of plans (available in specific libraries) composed of several actions. Jason's agents are programmed based on the definition of objectives, intentions, beliefs, plans and actions internal to the agent and actions performed in the environment. An SMA in Jason does not traditionally have an interface for capturing perceptions

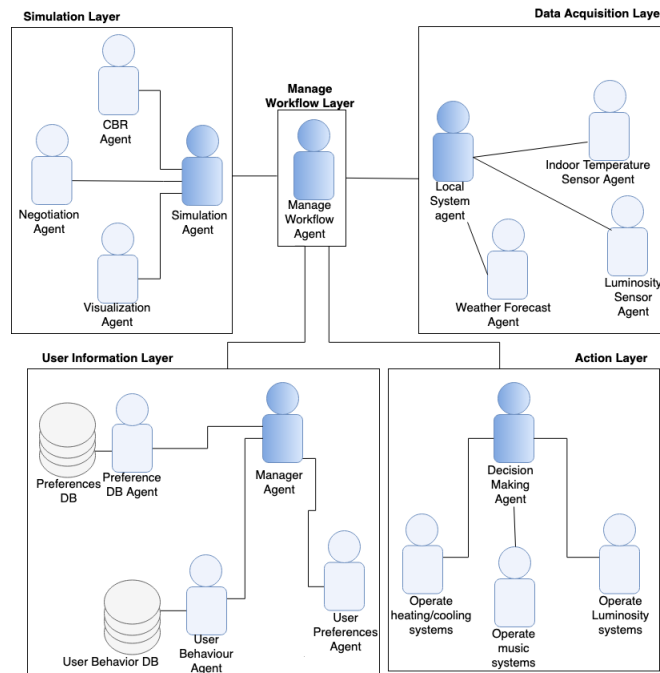


Fig. 7: Architecture of the multi-agent system

directly from the real world using sensors. The ARGO custom architecture is used for this, Jason uses a simulated environment.

### 3.2 The ARGO Architecture

ARGO is a customized Jason agent architecture to enable the programming of robotic and ubiquitous agents using different prototyping platforms.

ARGO allows intermediation between cognitive agents and a real environment (using controllers) through *Javino* middleware, which communicates with the hardware (sensors and actuators). In addition, as the use of BDI on robotic platforms can generate bottlenecks in the processing of perceptions and, consequently, unwanted delays in execution, this extension also has a perception filtering mechanism at run time [13].

An SMA using Jason and ARGO can be made up of traditional Jason agents and ARGO agents that work simultaneously. Jason agents can carry out plans and actions only at the software level and communicate with other agents in the system (including ARGO agents).

An ARGO agent, on the other hand, is a traditional agent with additional characteristics, such as, for instance, the ability to communicate with the physical environment, perceive it, modify it and filter the perceived information.

Figure 7 represent the architecture separation into different layers, to be easily identified the purpose of each, and agents containing it.

The layers description is as follows:

- **Data acquisition layer**, which will import the necessary information for the agents operation, namely information of interior and exterior temperature and light sensors.
- **User layer**, in this layer we will have an agent that will represent each user and his preferences that must be taken into account in the negotiation process.
- **Local System layer**, here each local system will be represented by an agent, which contains all the information necessary to this location, either the referred to user preferences, or local/users security (maximum/minimum temperature, safety values for CO<sub>2</sub>, etc.).
- **Simulation layer**, in this layer will be the negotiation between the different agents involved, namely the management of conflicts between the different users and local systems. After the negotiation process ends we will have as result the values to apply in the place.
- **Action layer**, after the process is executed in the simulation layer, and the values to be applied are obtained. These values are used in this layer and sent to the actuators that will apply them in the different automation systems present the local.

## 4 Discussion and Conclusions

With this work, the total development of an architecture and respective cognitive model for a Smart Home was achieved, using an SMA with BDI agents, developed using Jason and ARGO.

The main objective of this work was mainly to verify the potential that this type of architecture has for the development of ubiquitous SMA using low cost hardware, such as Raspberry's.

This being a feasible proposal for the problem to be solved in this project, thus being a viable solution and that solves all the constraints adjacent to the problem.

The agent system modeling is fully developed. At this stage the agent layer is developed and implemented, and is now in a testing phase in the testing environment developed for this project.

For future work, the results of the testing phase, will be analyzed and evaluated and with that results improve this project and support other works in this field.

### Acknowledgments

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.



## References

1. Andrade, J.P.B., Oliveira, M., Gonçalves, E.J.T., Maia, M.E.F.: Uma abordagem com sistemas multiagentes para controle autônomo de casas inteligentes. XIII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC) (2016)
2. Benta, K.I., Hoszu, A., Văcariu, L., Creț, O.: Agent based smart house platform with affective control. In: Proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to increase Digital Citizenship. pp. 1–7 (2009)
3. Bluetooth, S.: Bluetooth core specification version 4.0. Specification of the Bluetooth System (2010)
4. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason, vol. 8. John Wiley & Sons (2007)
5. Camps-Mur, D., Garcia-Saavedra, A., Serrano, P.: Device-to-device communications with wi-fi direct: overview and experimentation. *Wireless Communications, IEEE* 20(3), 96–104 (2013)
6. Chaouche, A.C., Seghrouchni, A.E.F., Ilié, J.M., Saidouni, D.E.: A higher-order agent model with contextual planning management for ambient systems. In: Transactions on Computational Collective Intelligence XVI, pp. 146–169. Springer (2014)
7. Kazanavicius, E., Kazanavicius, V., Ostaseviciute, L.: Agent-based framework for embedded systems development in smart environments. In: Proceedings of the 15th International Conference on Information and Software Technologies IT. pp. 194–200 (2009)
8. Martins, R., Meneguzzi, F.: A smart home model to demand side management. In: Workshop on Collaborative Online Organizations (COOS'13)@ AAMAS (2013)
9. Martins, R., Meneguzzi, F.: A smart home model using jacamo framework. In: 2014 12th IEEE International Conference on Industrial Informatics (INDIN). pp. 94–99. IEEE (2014)
10. Oliveira, P., Matos, P., Novais, P.: Behaviour analysis in smart spaces. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld). pp. 880–887. IEEE (2016)
11. Oliveira, P., Novais, P., Matos, P.: Challenges in smart spaces: Aware of users, preferences, behaviours and habits. In: International Conference on Practical Applications of Agents and Multi-Agent Systems. pp. 268–271. Springer (2017)
12. Oliveira, P., Pedrosa, T., Novais, P., Matos, P.: Towards to secure an iot adaptive environment system. In: International Symposium on Distributed Computing and Artificial Intelligence. pp. 349–352. Springer (2018)
13. Stabile, M.F., Sichman, J.S.: Evaluating perception filters in bdi jason agents. In: 2015 Brazilian Conference on Intelligent Systems (BRACIS). pp. 116–121. IEEE (2015)
14. Want, R.: Near field communication. *IEEE Pervasive Computing* 0(3), 4–7 (2011)
15. Wooldridge, M.: An introduction to multiagent systems. John Wiley & Sons (2009)