Specification of a Smart Classroom based on Agent Communities

Jose Aguilar[1], Luis Chamba-Eras, Jorge Cordero[2]

[1] Universidad de Los Andes, Mérida, Venezuela
Prometeo Researcher, Universidad Técnica Particular de Loja, Loja, Ecuador
aguilar@ula.ve
[2] Universidad Técnica Particular de Loja, Loja, Ecuador
{lachamba6,jmcordero}@utpl.edu.ec

Abstract. For the development of distributed applications, it is required to define a formalization of the process of implementation. Particularly, we are interested in one type of Ambient Intelligence (AmI), the Smart Classroom. In this paper we propose the implementation of a Smart Classroom, called SaCI, using the concept of communities of agents. With this concept, we carry out the definition and implementation of sets of agents according to their roles, functionalities, characteristics, among others, in SaCI. Each community can be designed and implemented independently and later be integrated in SaCI. In this paper we present this approach and its implementation in SaCI.

Keywords: smart educational environment, multi-agent system, ambient intelligence.

1 Introduction

Ambient Intelligence (AmI), is an area of Computing Sciences dedicated to design spaces that are composed by smart or not objects which interact with each other, in order to define applications to support users of these, in their activities. AmI is based on new areas in Computing Sciences, especially in Ubiquitous Computing. Among its features are the ability to detect environmental information, based on the interaction of autonomous devices with computing capabilities, and reasoning from data, information and knowledge accumulated, in order to select actions to execute. There are many applications: smart home, services in the health sector, among others.

In this paper, our interest is in education, particularly in the Smart Classroom. A Smart Classroom is an area where the Ubiquitous Technology helps the learning process in a transparent manner, incorporating new ideas and approaches in the educational process [4, 5], [8, 9]. In previous work, we have specified a Smart Classroom called SaCI [6, 7], [10]. SaCI consists of devices and software, primarily educational (e.g., smart boards, Virtual Learning Environments (VLE), Intelligent Tutoring Systems (ITS), among others), which it adapts and integrates to the course according to the necessities of the students. This adaptation of the different components of SaCI is possible due to its autonomic and reflective capacities.

The works [6, 7], [10] define the conceptual design of SaCI, with the details of each component. This conceptual definition of SaCI needs to be implemented in a

Á. Rocha et al. (eds.), *New Advances in Information Systems and Technologies*,
Advances in Intelligent Systems and Computing 444,
DOI 10.1007/978-3-319-31232-3_95

computational platform. This paper has this goal, it defines the procedure to develop SaCI computationally, and presents examples of this implementation.

Particularly, we propose to use the concept of communities of agents in order to guide the implementation process. A community of agents is a set of agents, with the same roles, functionalities or characteristics in SaCI. They can be developed independently of the rest of the communities, for later integration. In this paper we propose this approach to implement SaCI, for that, we have defined the communities to be developed in SaCI, we show the development of two of them, and carry out some tests in the implemented platform, to test SaCI.

2 Smart Classroom SaCI

SaCI is a Smart Classroom proposed in [10], where its deployment environment (middleware) was proposed in [6, 7] (called AmICL). SaCI proposes a smart, student-centered classroom, which supports the learning process, through collaborative devices and applications that facilitate self-training. To do this, SaCI has different types of components: hardware (such as smart boards, projectors, cameras, etc.) and software (ITS, VLE, among others). In [10] has been proposed the SaCI model based on the paradigm of Multi-agent Systems (MAS). The reflective middleware for smart learning environments [6, 7] is shown in Fig. 1.

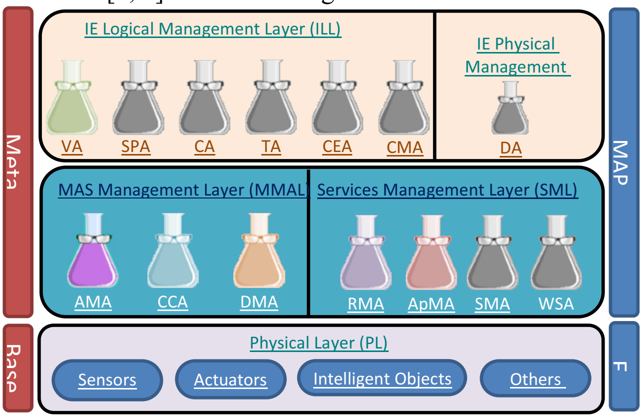

Fig. 1. AmICL architecture.

AmICL has five levels. A physical layer with components that interconnect the elements of an AmI (software or hardware), such as APIs, libraries, etc. That level works directly with the operating system. The SMA management level consists of the classic multi-agent community, defined by FIPA to support the implementation of multi-agent applications [2]. The service management layer has the responsibility to seek the required services in the cloud, particularly educational. The AmI physical layer represents the various devices in the environment, represented as agents. The AmI logical layer (ILL) represents the different software applications used in the educational platform, it is also represented as agents. In particular, this layer defines the applications (software) and individuals present in SaCI as agents, which contain metadata that defines them. The last two layers are those that define SaCI. Finally, in the real classroom of AmICL is where the devices (sensors, smart cameras, etc.) and software (VLE, etc.) of SaCI are deployed.

In specific, the agents of the middleware are, in MMAL: Agent Management of Agents (AMA), CCA (Communication Control Agent) and DMA (Data Management Agent, they are defined in [2]; SML is defined by the Services Management Agent (SMA), the Web Service Agent (WSA), the RMA (Resource Management Agent) and the ApMA (Applications Management Agent). The agents of ILL are: the Vision Agent (VA), the Student's Profile Agent (SPA), the Collaborative Agent (CA), the Tutor Agent (TA), the Content Management System Agent (CMA), and the Collaborative Environments Agent (CEA). Finally, the IE Physical Management Layer is composed of a single type of agent, the Device Agent (DA).

The SaCI can take advantage of cloud computing. In [6, 7] is described AmICL, as the reflective middleware based in Cloud Learning, which exploits the academic services on the cloud. In particular, the educational services in the cloud are to enhance the learning experience of students. With AmICL, the SaCI avoids any need to locally maintain complex computing infrastructures, and exploits the "as-a-Service" paradigm [12]: Database-as-a-Service, Security-as-a-Service, Sensing and Actuation-as-a-Service, Cloud-Based Analytics-as-a-Service (CLAaaS), Data Mining-as-a-Service (DMaaS), among others. For example, it can use cloud environments like CLAaaS and DMaaS, to obtain services for specific tasks about social learning analytic, which have Big Data challenges [11].

3 Specification of SaCI based on communities of Agents

In this paper, we define SaCI based on the notion of communities of agents, where the agents of SaCI are grouped according to the similarities at the level of characteristics, functionalities or roles in SaCI. In this way, we propose the following communities:

- Community of Agents that manage the Ambient Conditions in SaCI
- Community of Agents that manage the Learning Resources
- Community of Agents that manage the Learning Process
- Community of Agents that represent the Human Beings
- Community of Agents that represent the Mobile Objects (e.g. Robots) in SaCI
- Community of Agents that represent Static Objects in SaCI

According to these communities, we can develop each set of these types of agents of SaCI independently. Now, we define these communities using MASINA [1].

Community of Agents that manage the Ambient Conditions in SaCI

These sets of agents have the task of being managers of SaCI environmental conditions such as temperature, lighting, noise, among others. In particular, they are focused on monitoring and control of the temperature, noise, and lighting in the SaCI. Particularly, we have defined two agents:

- The temperature agent: determines and regulates the temperature in the SaCI.
- The lighting agent: regulates the light in the SaCI.

They adapt these variables to work in an average of the necessities of the people in a SaCI. Table 1 shows the description of one of these agents (for the rest see [3]).

Table 1. **Temperature agent description.**

Name: temperature agent
Position: Community of Agents that manage the Ambient Conditions
Description: This agent determines and regulates the temperature in SaCI

Community of Agents that manage the Learning Resources

This community aims to facilitate the learning processes through a set of agents, which search learning resources required by the students. The next agents compose it:

- Application Manager Agent: it receives the requirement of information from the rest of communities. It creates a recommender agent by request.
- Recommender agent: according to the student profile, it searches the learning resources on the local repository, and if it doesn't exist, it searches in Internet.
- Repository agent: it manages the local repository of educational resources.

Table 2 shows the description of one of these agents (the rest in [3]).

Table 2. Recommender agent description.

Name: Recommender agent
Position: Community of Agents that manage the Learning Resources
Description: This agent searches learning resources according to the student profiles

Community of Agents that manage the Learning Process

This community defines the applications that enable the creation and management of teaching and learning spaces through Internet, where teachers and students can interact during the training process. The main component is the VLE agent, which allows the managing of a course. The VLE agent presents resources and activities within a course; for its different phases. Some of the functionalities of this agent are:

- Content management, creation, storage, and access to learning resources.
- Mapping and planning of the curriculum, lesson planning, etc.
- Student management: student information, etc.
- Space of communication and collaboration - emails, notices, chat, wikis, blogs.

Other applications in this community are the ITS, the educational content management systems, among others. Table 3 shows the VLE agent description.

Table 3. VLE agent description.

Name: VLE agent
Position: Community of Agents that manage the Learning Process
Description: Manages the learning process for a course

Community of Agents that represent the Human Beings

This community aims to represent teachers and students in the SaCI. It has an abstraction of each Human Being in the SaCI, representing their capabilities, skills, learning style (for the students), etc. Table 4 shows the description of this agent.

Table 4. Virtual Student agent description.

Name: Virtual Student agent
Position: Community of Agents that represent the Human Beings
Description: This agent is an abstract of the virtual students, with its style of learning, etc. Each virtual student has a virtual agent in SaCI

Community of Agents that represent the Mobile Objects in the SaCI
This community represents the different Mobile Objects (e.g., Robots) in SaCI like agents. Table 5 shows the general description of this agent.

Table 5. Robot agent description.

Name: Robot agent
Position: Community of Agents that represent the Mobile Objects
Description: This agent is the abstract of the Robot in a SaCI

Community of Agents that represent Static Objects in a SaCI
This community defines the different static objects in SaCI like agents. Some examples of static agents are the smart boards, the cameras, etc. Table 6 shows the description of one of these agents (for the rest see [3]).

Table 6. Smart Board agent description.

Name: Smart Board agent
Position: Community of Agents that represent Static Objects
Description: This agent is the abstract of the smart board in a SaCI

4 Implementation

In this section we describe the implementation of some of the communities, for the rest see [3]. We have used JADE for the implementation of the agents in the SaCI.

Community of Agents that manages the Ambient Conditions in a SaCI
We have defined different agents in this community, one for each sensor in an AmI. Each agent has one sensor to monitor AmI, and a controller (e.g., the temperature (ventilators), light (blinds), etc. Fig. 2 shows the interface between these agents and the SaCI (the detailed implementations of these agents with JADE are in [3]).

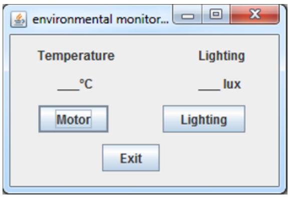

Fig. 2. Interface of the temperature and light agents

Community of Agents that manage the Learning Resources
The main agent in this community is the Recommender Agent. This agent has been developing using an API to call to MERLOT. MERLOT is an application to find online learning materials. With MERLOT, we can access Learning Object Repositories (the detailed implementation of this agent with JADE is in [3]). The Recommender Agent prepares the query and calls MERLOT to search for the information required (see Fig. 3). The sets of agents in this community are:

- Request management agent: manages the requests for information. Many managers will be created, as requests exist.

- Recommender agent: requests learning resources according to the student profile data and the learning component.
- Repository management agent: manages the consultations to the repository of educational resources.

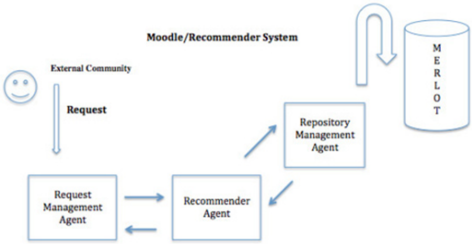

Fig. 3. Architecture of the recommender agent

Community of Agents that manage the Learning Process

The main agent is the VLE Agent. In our case it is defined by MOODLE, a VLE widely used. Fig. 4 shows our VLE agent like an abstraction of MOODLE, with additional capabilities to store information about the context. In general, users request resources they need in a given context, the VLE responds after consulting the database of resources, with the resources that may be useful to users. All interactions are stored in the database of academic management system (SGA) (the detailed implementations of these agents with JADE are in [3]).

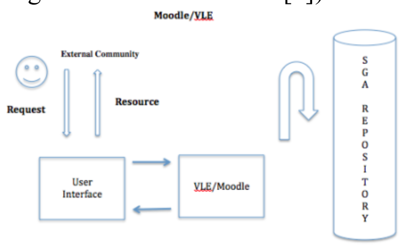

Fig. 4. Architecture of the interaction with VLE

Community of Agents that represent the Mobile Objects (e.g. Robots) in the SaCI

Each mobile object has one agent. To define each robot, we need to use the specific library or API of each Robot. The main aspect is to connect the robot with the agent implemented in JADE. For that, we used FIPA-ACL messages, which allow sending actions and messages between the agent implemented in JADE and the robots (see Fig. 5). Two of these possible actions are move and power-up (the detailed implementations of these agents with JADE are in [3]).

Community of Agents that represent Static Objects in a SaCI

The main agent in this community is the Smart Board Agent (the detailed implementation of this agent with JADE is in [3]). This agent uses the library of the Smart Board to communicate with it (see Fig. 6). The main function of this agent is to receive requests from other communities to present open educational resources, to develop dynamic multimedia activities between teachers and students, among others. SaCI exploits the user interface of the Smart Board to show the learning resources, to allow the interaction of activities between human beings, etc.

```
Application app = new Application(args);
Session session = new Session();
Future<Void> fut =session.connect
("tcp://172.17.41.159:9559");
fut.get();
com.aldebaran.qimessaging.Object tts = null;
tts = session.service("ALTextToSpeech");

boolean ping = tts.<Boolean>call("ping").get();
if (!ping) {
    System.out.println("Could not ping TTS");
} else {
    System.out.println("Ping ok");
}
tts.call("say", "Hola mundo, soy Nao");

Connection objConnection = new Connection();
myAgent.addBehaviour(objConnection);
```

Fig. 5. Code to implement the interface between the Robots and the agents in JADE

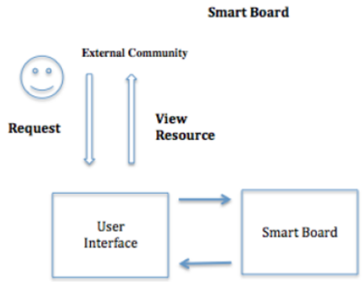

Fig. 6. Smart Board in SaCI

5 Experiments

SaCI has six main conversations in order to adapt the learning process to the needs of the students. The conversations are:

- Online tutoring process
- Set-up environment variables
- Stimulating the learning process
- Teacher´s guides to the learning process
- Feedback processes

In this section, we explain only some of them.

Online tutoring process

The online tutoring process (see conversation in Fig. 7) begins identifying users that enter to the VLE. Once VLE has the UID of the different users of this session, it instantiates the corresponding agents for each user. The AmICL determines that it must prepare the smart board available in the Intelligent Classroom. Thus, TA locates the course data and plans the class; with that information, it locates today class' slides through a storage service (in the course plan, it is indicates where the slides are stored, and in this way it can call the RMA).

The slides file (managed through the RMA) is sent to the smart board, then the DA (Smart Board) is prepared to begin class when teacher ordered. If the user is a student, the academic data is retrieved to determine the learning style and usage history of the environment for the group of students in the AmI. For that, VLE invokes a *clustering service* to define the groups of students (each group is a pattern). With this data, VLE asks the recommender system (it is inside of the RMA) for activities, digital contents, etc. to each group of students. In this case, each group is a pattern or style of learning to be exploited by the recommender system, to search more accurate information. That is, the clustering process allows an intelligent search (that is carried out by the recommender system) of learning resources, which are shown into the environment by the smart board according to the planning defined by VLE. Then, the students interact with these learning resources via the smart board, and TA monitors the work of the students. This is a cyclical process that is done in each tutoring session. At the end, the VLE establishes a student score (evaluation), and updates the learning profile of the students in function of these results (learn). Additionally, the VLE establishes an online tutoring model of the process to analyze the elements (chat, email, etc.), activities, etc., used during the section. For that, it invokes a *data mining service*, which is going to determine a descriptive model using the respecting agents of SML.

Set-up Environment Variables
The agents of the Community of Agents that manage the Ambient Conditions in SaCI monitor the environment, and can change when a robot agent require it or when they determine in the cloud a change of the environmental variables which are outside of the values allowed. This action of control is carried out by the actuators of the agents (see conversation in Fig. 8).

5 Conclusions

In this work, we have shown the implementation of SaCI. For that, we have introduced a concept, called communities of agent, in order to ease the process of development of the agents. This approach allows defining each community, and arriving at a prototype, in order to test them in an isolated environment. This allows for independent development, and only when the functionality of each community is correct is integrated.

In general, the JADE agent platform simplifies the implementation of SaCI [3], and the MASINA methodology the design of SaCI, but they are not enough for an efficient development of the agents. We need an approach to guide the development process, this should determine how to proceed with the specification of the agents. The "communities of agent" concept allows for a defining group of agents with similar tasks, with similar characteristics and common roles in SaCI. This allows developers and professionals specialize in specific aspects of SaCI. This approach gives a large scalability to SaCI, because new communities can be incorporated, and before to be integrated, they can be tested.

Acknowledgments. Dr Aguilar has been partially supported by the Prometeo Project of the Ministry of Higher Education, Science, Technology and Innovation of the Republic of Ecuador. This work has been partially supported by the UTPL Project entitled: "Medios de Gestión de Servicios (Middleware) Inteligentes para Entornos de Aprendizaje Virtual".

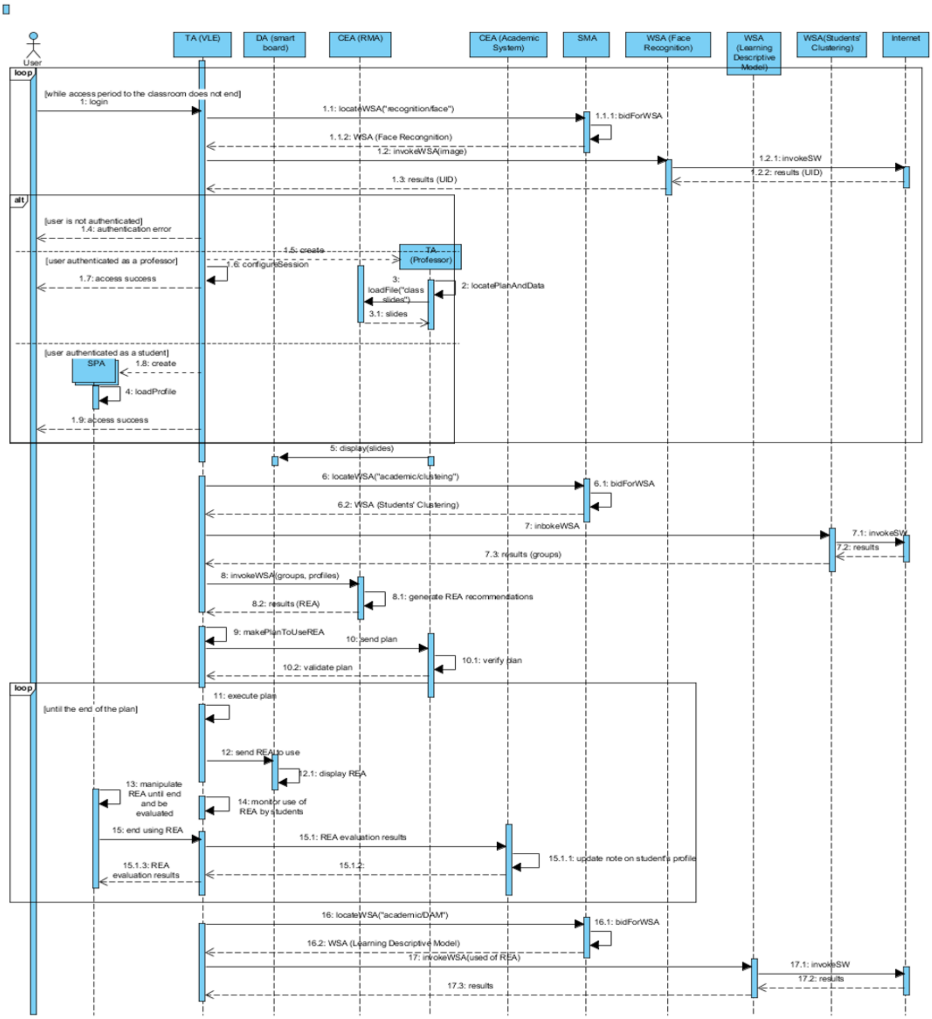

Fig. 7. Conversation "Online tutoring process".

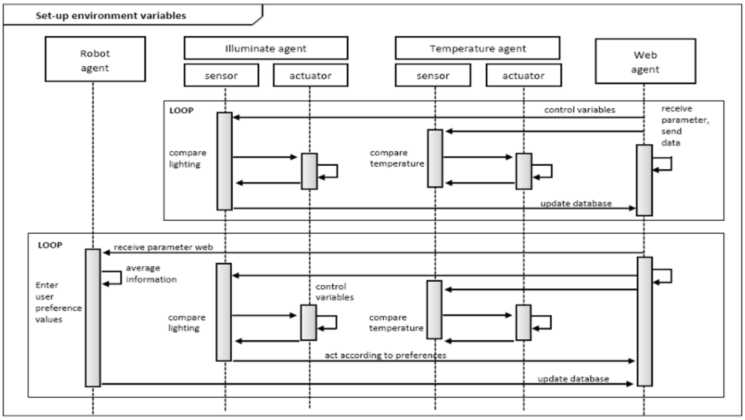

Fig. 8. Conversation "Set-up environment variables".

References

1. Aguilar, J., Besembel, I., Cerrada, M., Hidrobo, F., Narciso, F.: Una metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes. Revista Iberoamericana de Inteligencia Artificial, vol. 12, no. 38, 39–60 (2008)
2. Aguilar, J., Ríos, A., Hidrobo, F., Cerrada, M.: Sistemas Multiagentes y sus aplicaciones en Automatización Industrial. Universidad de los Andes, Caracas (2013)
3. Aguilar, J., Cordero, J., Chamba-Eras, L.: The details to specify SaCI. Technical report, Universidad Técnica Particular de Loja (2015)
4. Mhiri, F., Ratté, S.: AARTIC: development of an intelligent environment for human learning. ACM Sigcse Bulletin, vol 41, no. 3, 359 (2009)
5. Mikulecký, P.: Smart environments for smart learning. In 9th International Scientific Conference on Distance Learning in Applied Informatics DIVAI, pp. 213-222, Sturovo (2012)
6. Sánchez, M., Aguilar, J., Valdiviezo, P., Cordero, J.: Basic features of a Reflective Middleware for Intelligent Learning Environment in the Cloud. In Asia-Pacific Conference on Computer Aided System Engineering, pp. 1-6. Quito (2015)
7. Sánchez, M., Aguilar, J., Cordero, J., Valdiviezo, P.: A Smart Learning Environment based on Cloud Learning. International Journal of Advanced Information Science and Technology, vol. 39, no. 39, 39-52 (2015)
8. Shi, Y., Qin, W., Suo, Y., Xiao, X.: Smart classroom: Bringing pervasive computing into distance learning. In Handbook of Ambient Intelligence and Smart Environments, pp. 881–910. Springer, USA (2010)
9. Stenvall-Virtanen, S., Nordell, K.: Smart Environments: Technology, Protocols and Applications. Technical report, University of Turku (2014)
10. Valdiviezo, P., Cordero, J., Aguilar, J., Sanchez, M.: Conceptual Design of a Smart Classroom Based on Multiagent Systems. In ICAI-International Conference on Artificial Intelligence, pp. 471-477. Las Vegas (2015)
11. Sharma, S., Tim, U., Gadia, S., Wong, J.: Proliferating Cloud Density through Big Data Ecosystem, Novel XCLOUDX Classification and Emergence of as-a-Service Era Download. Technical Report, Iowa State University, USA (2015)
12. Sharma, S. Evolution of as-a-Service Era in Cloud. Technical Report, Iowa State University, USA (2015)