

JADE Multi-Agent Middleware Applied to Contribute to Certificate Management of Students

Fatiha Aityacine*, Badr Hssina, Belaid Bouikhalene

Sultan MoulaySlimane University, Faculty of sciences and technology, Beni Mellal, Morocco

*Corresponding author, e-mail: fatiha.aityacine@yahoo.de

Abstract

In this article, we present a multi-agent approach that aims to design, modeling and implementation of an application "smart school". Indeed Several institutions adopt the computerized management of education to meet the needs of students using multi-agent systems. They have the ability to act simultaneously in a shared environment. The purpose of this approach is to automate some administrative services of education, based on the theory of distributed artificial intelligence (DAI) and multi-agent systems (MAS). This multi-agent application integrates entities called agents that cooperate and communicate them to perform specific tasks. Our system is based on the middleware JADE (Java Agent DEvelopment Framework) used for the implementation and agents management. This model based on multi-agent systems is tested on the personal data of an experiment conducted with the students of Sultan MoulaySlimane University in BeniMellal.

Keywords: SMA, web agent, JADE, school management

Copyright © 2015 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Nowadays computerized management is becoming more and more common because it helps simplify tasks and procedures in companies and institutions. The educational field is no exception to this trend. Indeed Several institutions adopt the computerized management of the school to meet the students' needs accurately and efficiently. These classic procedures are long and expensive [1] compared to the time and effort of students and administrators [2].

The procedures and problems of education services discussed in this article are:

- a) Need to file the application manually in the registrar's office (Moving studying)
- b) The availability of staff is limited.
- c) Need staff to processing applications.
- d) Risk of loss forms (paper) if the numbers of these become huge.
- e) Control of the number of documents (certificate and / or transcripts) from each student for a year.

For these reasons and the other concerning the enrichment of information of our university system. We thought automate some schooling services. Our goal is to model and develop a distributed application based on intelligent agents[3] whose goal is to properly meet these requirements. Our multi-agent system is based on entities called agents communicating with each other to coordinate their intelligent and cooperative behavior to to achieve the overall objective of the system.

Multi-agent systems (MAS) [4] are systems that are based on knowledge sharing. It is based on the complex interaction patterns that result in solving strategies such as cooperation, coordination, negotiation [5].

There are currently several agents platform, which facilitate the development and handling of multi-agent systems. The choice of agent platform depends heavily on the the objective of application in question.

As part of our work, the platform should meet the following requirements:

- a) Because our multi-agent system is distributed, the selected platform must provide the necessary facilities and support for this feature.
- b) The agents needed are interactive and cooperative, hence the need to choose a platform allowing them to achieve their treatment without difficulties.

- c) The programming language and the nature of exchanged messages are also important. A fairly common language will be more useful.
- d) In addition, a rich platform in terms of documentation is easier to operate.

Based on the study and comparison of several platforms; MadKit (Multi-Agent Development Kit) [6], JADE (Java Agent DEvelopment) [7], GAMA (Gis& Agent-based Modeling Architecture) [8] and JADEX [7]. Our choice is focused on the JADE platform that better meets our criteria.

2. The Multi-Agent Platform JADE

The JADE platform (Java Agent Development Framework) [9] is an agent development environment fully implemented in JAVA. Its purpose is to simplify the development of multi-agent systems (MAS) while providing a complete set of services and agents naming service, yellow pages service, parsing service and protocol interaction library, ready to use. JADE The platform is based on the specification of FIPA (Foundation for Intelligent Physical Agent) [10] and has three main modules (required standards FIPA). These are activated at each start of the platform:

- a) **The Agent Management System (AMS)** : that agent is responsible for managing the activities of the platform, overseeing the creation and recording the removal of officers and their authentication, their access and use of system.
- b) **The intelligence agentDirector Facilitor (DF)** provides the service of "Yellow Pages" to the platform. It records the descriptions of agents and the services they offer. It must provide the most current information on registered agents in its repertoire to all other authorized agents.
- c) **The Agent Communication Channel (ACC)** handles communication between agents. It routes messages between agents located in the platform but also other agents reside on other platforms. ACC is the default communication medium that connects all agents on a platform.

Each instance of the JADE is called "Container" and may contain multiple agents. A set of containers are a platform. Each platform must contain a special container called main-container and all other containers register with this one from launch.

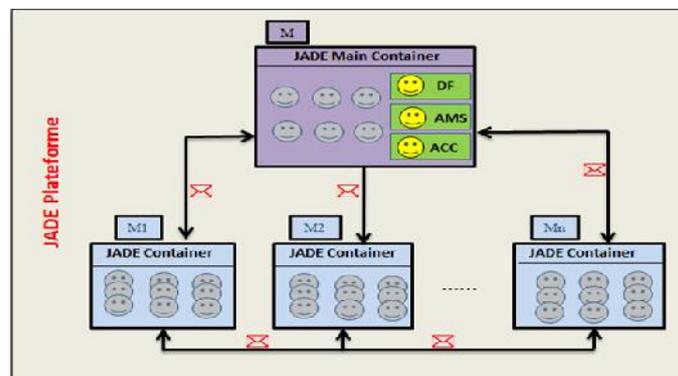


Figure 1. Architecture of the Jade platform

A main-container differs from other "simple" containers in that it always contains three special agents called AMS (Agent Management System), DF (Directory Facilitator) and ACC (Agent Communication Channel) [11] that are automatically launched at the launch of the main-container.

Another special agent to monitor communications between other agents is (Sniffer Agent): This agent specializes in monitoring streams of communication between agents.

There are in Jade a number of graphical tools that support the correction phase, usually very complex in distributed systems:

a) **The Dummy agent:** is a simple and very useful tool to visualize the exchange of messages between agents. The Dummy agent facilitates the validation of an agent prior to integration in the SMA and facilitates the "debug" if an agent fails.

b) **The Sniffer Agent:** keeps track of messages exchanged in a platform Jade. When the user chooses to monitor an agent or group of agents, each message directed to or from this agent, or group, is detected and shown in Sniffer window. The user can then view, save and load each message for later analysis.

c) **The Introspector agent:** to monitor and control the life cycle of a current officer and its exchanged messages (from the queue of messages sent and received).

3. Theoretical Study of our Application

In this part we will present the class diagram and the sequence of processes of an application for certification. Indeed the objective of this work is the realization of a distributed application for managing applications for certification within the education service using multi-agent JADE middleware.

3.1. The Class Diagram

The class diagram is composed of:

- The *Student* class: It contains the information of all students enrolled in the institution.
- Class *Request*: It stores all the requests made by students of the institution.
- The class *Status request*: It encompasses the statuses of each request accepted.
- Class *Certification*: It contains the types of certification that apply to students.
- Class of *study*: It contains the speciality that exist in the establishment.

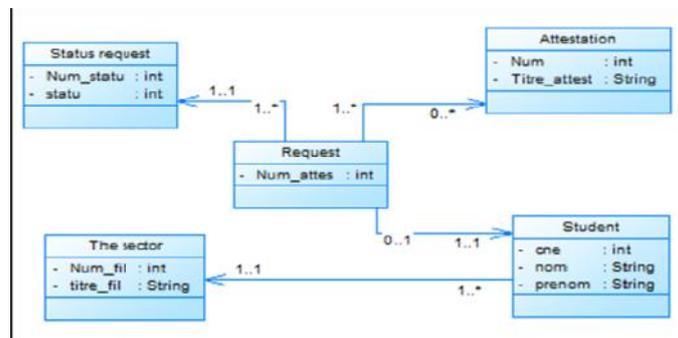


Figure 2. Class diagram of our application

3.2. The Unwinding Process a Request for Attestation

The **"Student"** agent connects to the platform, then it sends a ACL message to the agent **"Home"**, this message contains the **"CNE"** (National student code) and type of certification required. The **"Home"** agent receives these two items, and sends them to the **"Schooling Service"** agent. This must be a verification of the existence of that student in the database. If it does not exist, it returns a rejection message to the agent **"Home"**, the latter in turn, sends the same message to the agent **"Student"**.

The interactions diagram above Figure 3 shows all the stages through which a request for student upon connection to the platform until it is signed by the director of the institution with a precision tasks performed by each actor on our system.

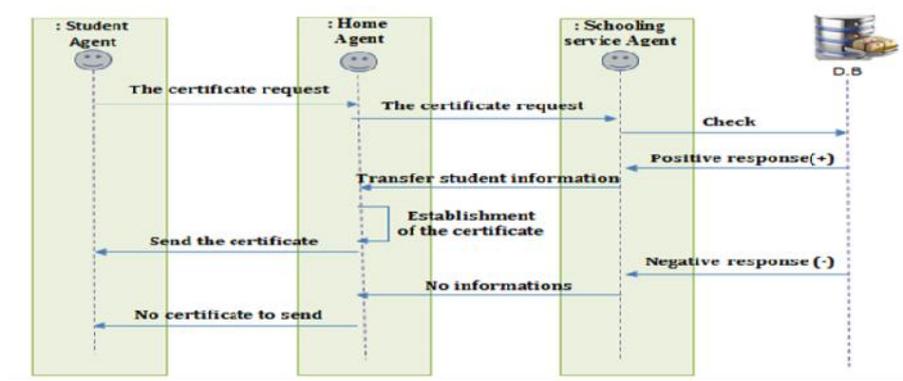


Figure 3. The diagram of interactions

4. Description of our Application

Our application is multi-agent. We created three agents (student Agent, Home Agent and Schooling service agent) each of these agents did work as we have already explained in the above diagram.

Firstly beginning with the Student Agent: Students can start the application from a smartphone that supports the Android system. The following interface is used to send a request containing a CNE and the type of certificate to the Home Agent.



Figure 4. Interface of the application

The home agent is the principal agent; this agent is deployed on a container on the central computer that contains the main-container.

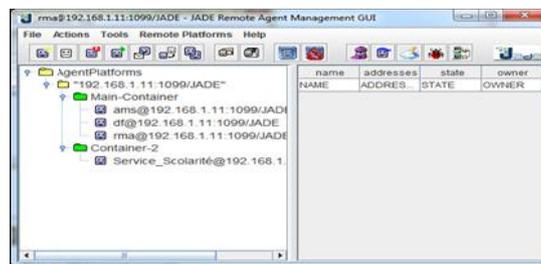
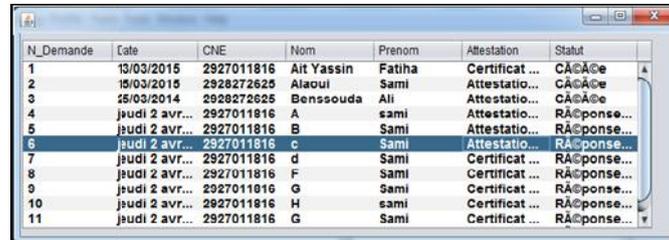


Figure 5. JADE Remote Agent Management GUI of our application

It is in this computer all other agents will be deployed because it contains the main-container. In another computer another agent has been programmed (school service) that makes the tasks we have already defined in the diagram of interactions. This agent uses a display interface of past interactions in the database and displays the status of requests every moment in an automatic way.



N_Demande	Date	CNE	Nom	Prenom	Attestation	Statut
1	13/03/2015	2927011816	Ait Yassin	Fatiha	Certificat ...	CA@À@e
2	16/03/2015	2928272625	Alaoui	Sami	Attestatio...	CA@À@e
3	26/03/2014	2928272625	Benssouda	Ali	Attestatio...	CA@À@e
4	jeudi 2 avr...	2927011816	A	sami	Attestatio...	RA@ponse...
5	jeudi 2 avr...	2927011816	B	Sami	Attestatio...	RA@ponse...
6	jeudi 2 avr...	2927011816	c	Sami	Attestatio...	RA@ponse...
7	jeudi 2 avr...	2927011816	d	Sami	Certificat ...	RA@ponse...
8	jeudi 2 avr...	2927011816	F	Sami	Certificat ...	RA@ponse...
9	jeudi 2 avr...	2927011816	G	Sami	Certificat ...	RA@ponse...
10	jeudi 2 avr...	2927011816	H	sami	Certificat ...	RA@ponse...
11	jeudi 2 avr...	2927011816	G	Sami	Certificat ...	RA@ponse...

Figure 6. JADE Remote Agent Management GUI of our application

After signing the request, the home agent generates an Excel file as the desired application by the student, treated 3 types of certification, each bearing the necessary information about the student (certificate of achievement, certificate registration and schooling certificate).



Figure 7. Certificate of Achievement generated

The agent "Schooling service" is made of successive and automatic updates to the request table in the database:

- Insertion in the table request after the verification "CNE";
- Changing the status "current = 2";
- Changing the status "Generate = 3";
- Changing the status "Signed = 4".

And every time it does a refresh of the request table to be updated.

Then, when the "status = Signed" agent "Schooling Service" sends the information of the student to the agent "Home". The latter creates the requested certificate as an Excel file. Then, it sends a message to the agent "Student" to tell him that the certificate is ready.

5. Conclusion

This article highlights the importance of using multi-agent approach to build a schooling management application. It also presents the specifications that we followed to develop multi-agent application using the JADE platform. The main objective of this application is to help students to have their schooling certificate.

References

- [1] ZHANG Xue-ling, MA Chao-qun. The Employment of Target Administration Theory in the Administration of Teaching Staff in Colleges and Universities. *Journal of Xianyang Teachers College*. 2005; 06.
- [2] Allen RJ, R Douence, D Garlan. *Specifying and Analyzing Dynamic Software Architectures*. In Proceedings of the 1998 Conference on Fundamental Approaches to Software Engineering (FASE98). Lisbon, Portugal. Springer, Berlin. 1998: 21-37.
- [3] Wooldridge M. *An Introduction to Multi Agent Systems*. Chichester, England: John Wiley & Sons. 2002.
- [4] A Halberstadt, L Mui. *Group and Reputation Modeling in Multi-Agent Systems*. Proc. Goddard/JPL Workshop on Radical Agents Concepts, NASA Goddard Space Flight Center. 2001.
- [5] BOURON T, FERBER J, SAMUEL F MAGES. *A Multi-Agent Testbed for Heterogeneous Agents*. In *Decentralized AI II*, Y. Demazeau et J.-P. Müller (Ed). NorthHolland. 1990.
- [6] Jorge J Gomez-Sanz, Rubén Fuentes, Juan Pavón, Ivan García-Magariño. *INGENIAS development kit: a visual multi-agent system development environment*. AAMAS '08 Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: demo papers. 2001: 1675-1676
- [7] Jade. Java Agent Development Framework. 2009.
- [8] Daniel G Brown, Rick Riolo, Derek T Robinson, Michael North, William Rand. Spatial process and data models: Toward integration of agent-based models and GIS. 2005; 7(1): 25-47.
- [9] G Caire, D Cabanillas. JADE tutorial: creating and using application specific ontologies. 2004.
- [10] FIPA00037. FIPA ACL communicative act library specification. Technical report, FIPA. 2002.
- [11] FIPA ACL Message Structure Specification.