

Implementation of an Adaptive Learning System Based on Agents and Web Services

<https://doi.org/10.3991/ijet.v15i14.13837>

Nihad El Ghouch ^(✉), El Mokhtar En-Naimi, Mohamed Kouissi
LIST laboratory, The Faculty of Sciences and Technologies, UAE,
Tangier, Morocco
nihad_elghouch@hotmail.fr

Abstract—Today, the integration of web services and agent technology into Internet applications has attracted the attention of many researchers, so that these applications allow a web service to call an agent service and vice versa. Web services are emerging and promising technologies for the development, deployment and integration of the Internet applications and the use of agents makes them dynamic and automatic, they can provide updates when there is new information available and improve the qualities of web services by exploiting the capacities and the characteristics of agents. In this context, we propose a prototype of a multi-agent adaptive learning system based on Incremental Hybrid Case Based Reasoning in order to support the learner in his learning process by offering him a learning path adapted to his profile and predict his future learning. This support will be achieved through the execution of a hybrid cycle of Case Based Reasoning which brings together a set of agents collaborating and interacting with each other to provide specific services.

Keywords—Adaptive Learning System, Multi-Agent System, Case Based Reasoning, Web Service, Java Agent DEvelopment platform.

1 Introduction

Since the appearance of the web, it was a huge warehouse of text and images, its evolution has made it also become a service provider. It was also a great success allowing simple interactions between human and machines. The exponential growth of the Web has contributed to the development of adaptive learning by expanding its possibilities in terms of interaction such as personalization. This growth has also enabled it to involve a considerable variety of technologies derived from artificial intelligence. This encourages the use of the Web and hypermedia in the transmission of knowledge in order to shape learning paths and guide the future development of learners.

Adaptive learning hypermedia systems allow to personalize the content and the learning path in an online learning environment to minimize learner disorientation [1], using methods and techniques to adapt content of these systems to the needs, interests, objectives, and individual traits of the learners [2].

This personalization requires individualized monitoring of the learners which plays an important role, because without information on the progress and the results obtained by the learners, it will be very difficult to support and control their behaviors. This monitoring requires "intelligent" components to detect in real time the learner's profile and learning preferences, in order to adapt automatically their learning at each step of the learning process.

To overcome the difficulty of real-time monitoring, we propose the implementation of an multi-agent adaptive learning system based on Incremental Hybrid Case Based Reasoning (IHCBR) in order to support the learner in real time during his learning process by offering him individualized learning according to his needs and preferences as well as predicting his future behavior, by integrating at each step of the hybrid cycle of Case Based Reasoning (CBR) [3] a set of agents to ensure a specific task and benefit advantages of agent technology.

To implement our architecture, we use the JADE platform (Java Agent DEvelopment) [4] for the implementation of our multi-agent system and the WSIG component (Web Service Integration Gateway) [5] as a gateway to offer discovery bidirectional and remote invocation of web services by JADE agents, as well as JADE agent services by web services.

In this paper, we introduce the concept of intelligent agent in the field of e-learning and we present an architecture of multi-agent adaptive learning system allows to follow the learner in real time and to predict his future learning. The rest of this paper is organized as follows: the second section presents the combination of web services and multi-agent systems. The third section is dedicated to presenting our architecture of a multi-agent adaptive learning system based on the IHCBR. In the fourth section, we present the implementation of our architecture by using the JADE platform and the WSIG component.

2 Web Service and Technology Agent

2.1 Web service

Web service refers to a self-contained, self-describing software component available on the Internet through a service provider, and accessible by customers via standard Internet protocols [6]. It takes up the principles of the web and applies them to the interactions between users and machines which are based on distributed architectures, which abandons centralized architectures (client / server).

Web services are implemented through technologies that facilitate description, discovery and communication between services, we quote: Universal Description, Discovery and Integration (UDDI) [7], Simple Object Access Protocol (SOAP) [8] and Web-Service Description Language (WSDL) [9]. Their objective is to promote a service-oriented architecture, incorporating heterogeneous, complex and distributed systems that prove to cooperate without resorting to specific integration. They allow to create new aspects of collaboration between remote applications.

2.2 Multi-agent system

Multi-Agent Systems (MAS) presents a new method for analyzing and implementing sophisticated applications because they are part of the field of distributed artificial intelligence [10]. A MAS is a system made up of a set of agents who are organized and work together to solve a common problem.

Today, most applications require distributing tasks between autonomous entities in order to achieve their objectives in an optimal manner. Since classical approaches are generally monolithic and their concept of intelligence is centralized, current applications are established using a multi-agent system.

2.3 Combination of web services and agents

The use of agents makes web services dynamic and automatic, they can provide updates when there is new information available and improve the quality of web services by exploiting the capacities and characteristics of agents.

Recently, we have noticed a rapprochement between MAS and Web services, this approach has attracted the attention of several researchers, we quote:

- The interconnection of application domains by allowing a web service to call an agent service and vice versa (inter-domain invocation)
- The use of a MAS as a proxy facilitating the localization of Web services [11]
- The use of a Web service as an architecture and a technology allowing to implement MAS accessible through the Web, whose agents offer their capacities via Web services [12, 13 and 14].

3 Our Proposed Approach

We propose a prototype of a multi-Agent adaptive learning system based on IHCBR which is based on the past experiences of other learners and the Felder-Silverman learning style model (FSLSM) [15], in order to create dynamically the personalized learning paths, to provide individualized monitoring in real time of each learner and to predict their future learning situation which changes dynamically over time. This prototype use a hybrid cycle of CBR which combines the classic cycle (the order of the steps is important, there is no change during the execution of the cycle) and the dynamic cycle (the order of the steps does not matter, a detected change results a re-execution of certain steps several times) and by integrating the concept of the agents.

Table 1 presents the distribution of agents in the different steps of IHCBR, as well as the tasks performed.

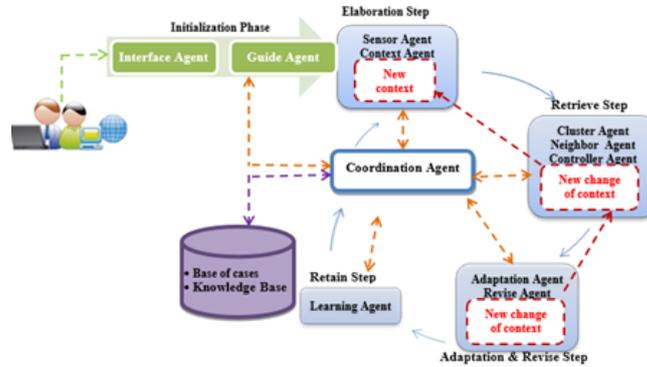


Fig. 1. Our proposed Architecture

Table 1. Agents of the system and their tasks in the different steps of IHCBR

Phase / Step		Agent	Tasks
Initialization phase		Interface Agent Guide Agent	Collect data from learners; Propose an initial learning path according to the FLSM profile.
Acquisition phase	Elaboration Step	Sensor Agent Context Agent	Collect learning traces; Make the right decision when faced with a learning problem
	Retrieve step	Cluster Agent Neighbor Agent Controller Agent	Create a list containing the experiences nearest to the current learning situation that can be modified following a change in the situation
	Reuse and Re- vise step	Adaptation Agent Revise Agent	Evaluate the resemblance between the selected past experiences and the current learning situation that can be modified following a change in the situation
	Retain step	Learning Agent	Report that the problem is resolved and that it becomes a new experience for future use.
Knowledge Base		Coordination Agent	Save and update all information sent by other agents; Ensure communication between the different agents.

3.1 Interface agent

The structure of the Interface agent is as follows:

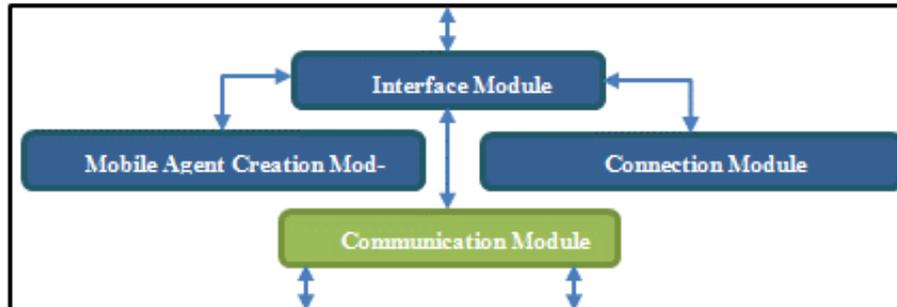


Fig. 2. Interface agent architecture

Interface module: This module is responsible for managing communications between the system and the connected learners.

Connection module: This module allows to:

- Authenticate the learners
- Register new learners
- Detect the profile of each learner (the learning style of FSLSM) [16].

Communication module : This module is responsible for sending and receiving messages from the Coordination agent. The Interface agent acquires all the requests sent by the learners and sends them to the Coordination agent, which in turn sends them to the other appropriate agents.

Mobile agent creation module : Through this module, the interface agent creates and dispatch mobile agents to learners' devices, if they are connected by mobile devices.

3.2 Guide agent

The structure of the Guide agent is as follows:

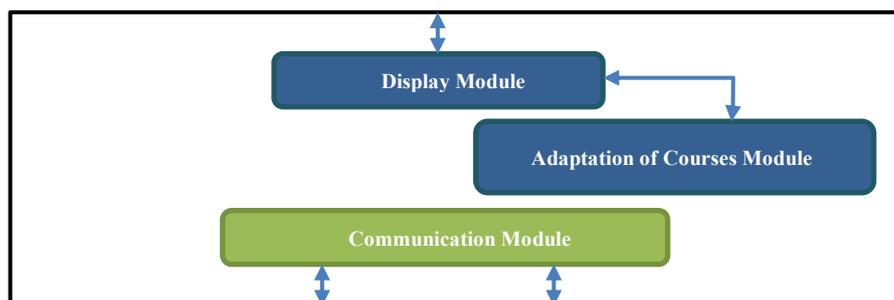


Fig. 3. Guide agent architecture

- a) **Display module:** He is responsible for the display and presentation of courses adapted to learners according to their profiles and real behaviors.

- b) **Communication module:** This module is responsible for sending and receiving messages from or to the other agents (Interface agent, Sensor agent and Coordination agent).
- c) **Adaptation of courses module:** This module is responsible for adapting the courses. This adaptation is carried out according to its learning profile by calculating the correspondence between the result of the FLSM learning test and the metadata describing the learning objects of the courses in order to start learning. At the start of the learning process, the adaptation is carried out according to his new context detected by the Sensor agent.

3.3 Sensor agent

The structure of the Sensor agent is as follows:

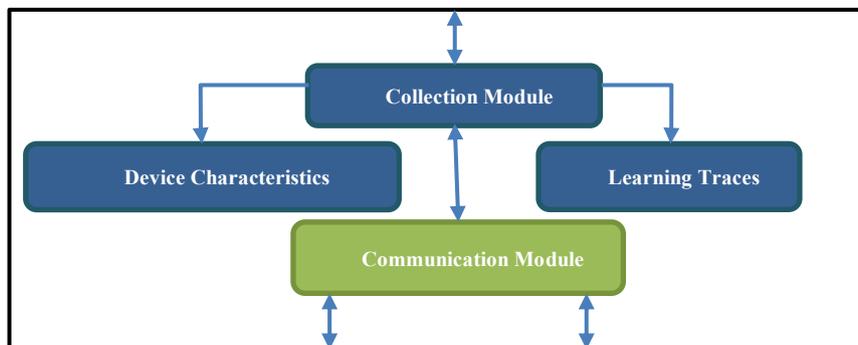


Fig. 4. Sensor agent architecture

- a) **Collection module:** This module captures the behavior of the learner during his learning process, in order to react automatically to the changes detected. The Sensor agent collects learning traces and characteristics of mobile devices at each instance t_i .
- b) **Communication module:** Through this module, the Sensor agent communicates with the Coordination agent in order to save learning traces and characteristics of mobile devices at each instance t_i of the learning process.

3.4 Context agent

The structure of the Context agent is as follows:

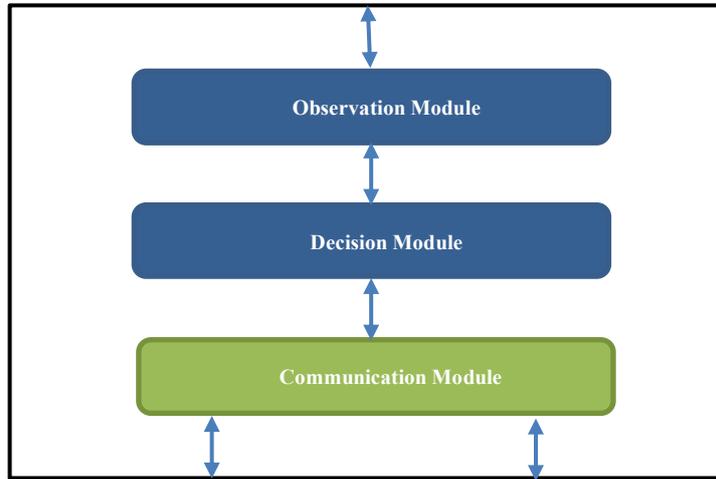


Fig. 5. Context agent architecture

- a) **Observation module:** This module makes it possible to control the learning process of the learner by comparing their initial learning path proposed by the Guide agent and their behavior observed at each instant t_i , in order to detect if the learner has encountered a learning problem.
- b) **Decision module:** At the level of this module, the Decision agent takes decisions when faced with a learning problem detected according to the observation module (the normal or abnormal learning process).
- c) **Communication module:** This module allows to send and receive messages from or to the Sensor agent and the Coordination agent.

3.5 Cluster agent

The structure of the Cluster agent is as follows:

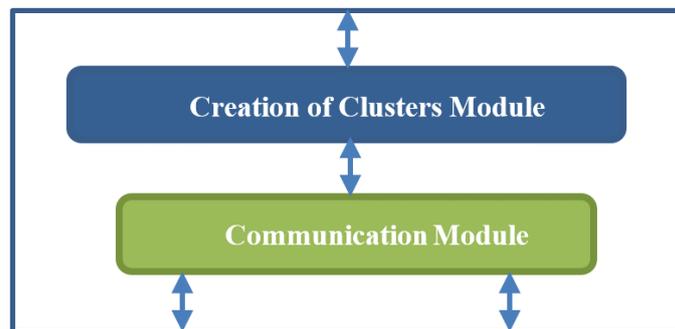


Fig. 6. Cluster agent architecture

- a) **Creation of clusters module:** The aim of this module is grouping and identifying the nearest cluster to the learner in difficulty. This cluster makes it possible to recover all the learners having similar behaviors to the current learning situation at each instant t_i in the form of a vector V_Clusti , using the Fuzzy C-Means method (FCM) [17].
- b) **Communication module:** This module allows the Cluster agent to communicate with the other agents in the system (the Context agent and the Coordination agent) by sending and receiving messages from or to these agents.

3.6 Neighbor agent

The structure of the Neighbor agent is as follows:

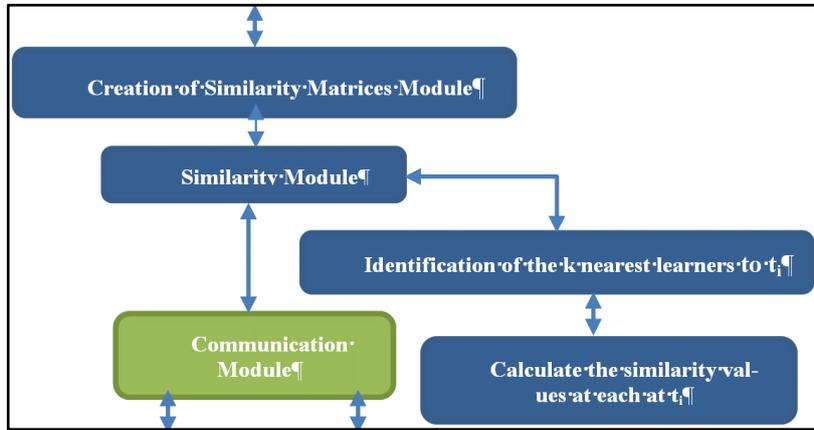


Fig. 7. Neighbor agent Architecture

- a) **Creation of similarity matrices module:** Each current or past learning situation (target case and source case) is described in the form of a vector describing the learning traces at each instant t_i , which are stored in the knowledge base (case base). The K- Nearest Neighbors method (K-NN) [18] makes it possible to find the k nearest learning situations of the current situation based on a similarity measure. In our case, the goal of using the K-NN method is to find the list of the k nearest learners at each instant t_i of the learning process in order to classify the majority learners who repeat several times and have optimal similarity. The list of the k nearest learners and the resemblance table will be update at each detection of a new change in the learning situation.
- b) **Communication module:** In order to retrieve the data necessary for the similarity calculation, the Neighbor agent needs to send and receive messages from or to the Coordination agent and the Cluster agent.

3.7 Controller agent

The Control Agent is listening for a new detected change. In this case, it sends a message to the Cluster agent which in turn sends it to the Neighbor agent to update the list of the nearest learners.

3.8 Adaptation agent

The Adaptation agent is responsible for validating the experiences of the nearest candidate learners, either by choosing the first experience offered, or by leaving the tutor the choice of choosing one from the experiences offered.

3.9 Revise agent

The Revise Agent is listening for a new detected change. In the case of the change, there will be a new elaboration of the current learning situation and an update of the list of nearest learners, which leads to a modification of the similarity table.

3.10 Learning agent

The Learning Agent is responsible for presenting the current learning situation and its solution as a new future experience to be saved in the knowledge base.

3.11 Coordination agent

The structure of the Coordination agent is as follows:

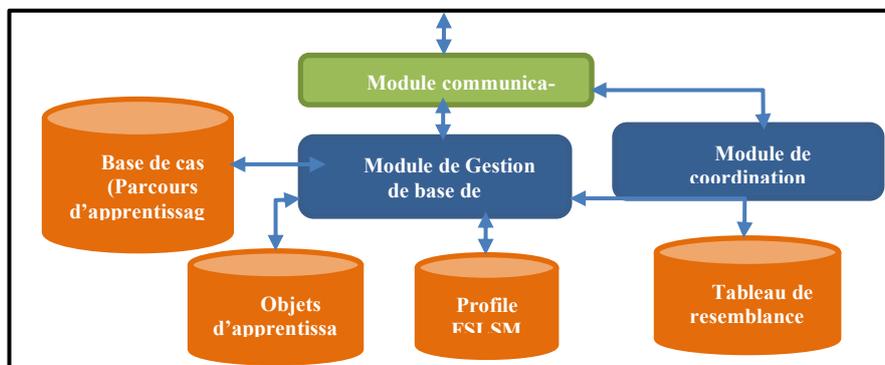


Fig. 8. Coordination agent Architecture

- a) **Coordination module:** In order for our system to function properly and gets better performance, it will be necessary to coordinate and control the different agents in the system. This module allows coordination and synchronization between messages sent and received between agents.
- b) **Database management module:** This module is responsible for the management and updating of the knowledge base (learner profiles, learning paths, learning objects). This module allows to manage:

- Requests of registration and identification;
 - Requests to save the learning traces and the characteristics of the mobile device at each instance t_i ;
 - Requests for updating profiles and learning paths of the learners.
- c) **Communication module:** This module allows the Coordination agent to communicate with other agents in the system, by sending and receiving messages from or to other agents.

4 Implementation of our Approach

4.1 Designing and modeling our approach (Reminder)

For more details, regarding the design and modeling of our approach, we invite you to review our work published in [19].

4.2 Basic architecture of JADE and WSIG

a) JADE platform

The best way to build a multi-agent system is to use a MAS platform allowing the commissioning of agents. There are several reasons for choosing the JADE platform [20]:

- It is compatible with most hardware and software configurations;
- It is generic for the design and execution of multi-agent systems;
- It makes it possible to develop MASs that comply with FIPA standards;
- It is implemented in JAVA language using a jar file;
- It has the ability to communicate between JADE agents and non-JADE agents.

The JADE platform uses:

- AID (Agent Identifier): this service makes it possible to distinguish and identify each agent, it builds from the name of the agent and the JADE container on which it runs;
- DF (Directory Facilitator): this service makes it possible to record the skills of each agent. The consultation of this service is used to obtain information on the skills of the agents in order to ensure good collaboration between them;
- AMS (Agent Management System): this service makes it possible to record the transport addresses of the various agents of the platform, by matching the agents with the AID in order to facilitate their control and supervision.

b) WSIG component

The WSIG presents a component of the JADE platform, allowing the translation of messages used by web services, to messages understandable by software agents. It uses a stack of standard web service protocols such as: UDDI, SOAP and WSDL.

WSIG is a web application composed of two main elements:

- WSIG servlet: presents the facet towards the internet
- WSIG agent: presents the gateway between the web and the agents.

4.3 Implementation steps

In order to implement the prototype of our multi-agent adaptive learning system, a set of steps to follow:

- a) **Creation of agents:** Each agent created is saved in a container which is also registered with a Main-Container.

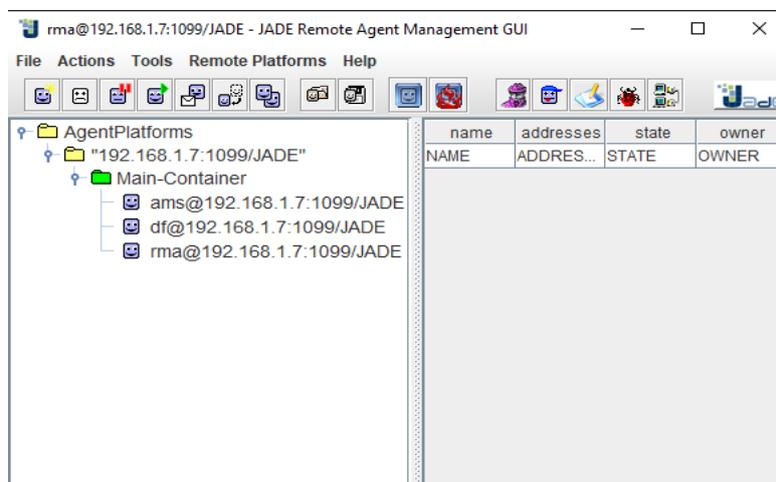


Fig. 9. Main Container OF JADE

As well as the use of WSIG allows to create an agent named "ControlWSIG-Container" (figure 10) allowing to integrate the agents existing in "Main Container" in the component WSIG.

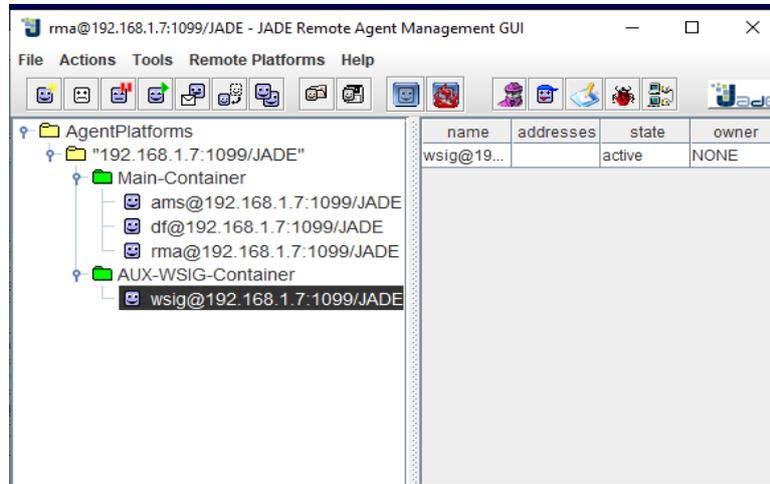


Fig. 10. Creation of the ControlWSIG Container

All agents in the system must inherit the JADE Agent class. This class redefines methods that define the agent's life cycle in the platform and implements the setup() method that will be called after the agent is instantiated by the container.

The following figure shows the registration of the Interface agent in the JADE platform in the container " Initialization":

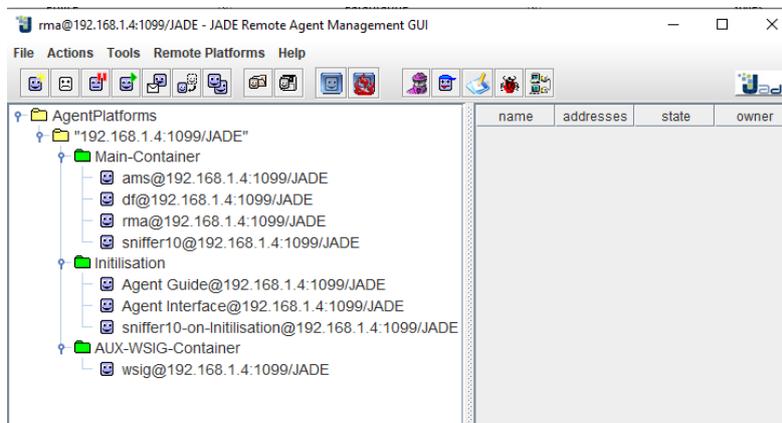


Fig. 11. Registering of the Interface Agent in JADE

To expose an agent service as a web service, we must set the wsig property to "true" in the properties of the service description when the DF registering:

```
ServiceDescription sd = new ServiceDescription ();
sd.addProperties (new property ("wsig", "true"));
```

In our architecture, we create six containers that contain the various agents listed in Table 1:

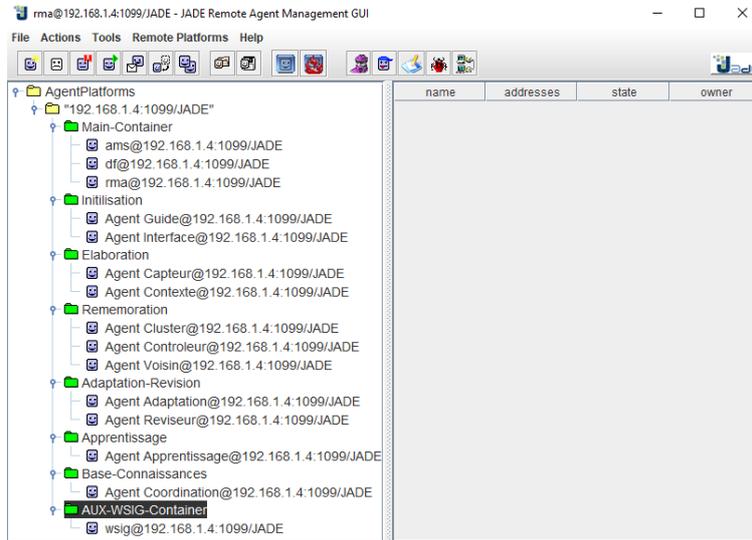


Fig. 12.Registration of different agents in our system

b) Registration oh the agents:

JADE agents publish their services in the DF, providing a structure called DF-Agent-Description. This last one contains one or more service descriptions describing a service provided by the registration agent. A service-description specifies one or more ontologies in order to access the published service, these ontologies define all the actions of the agent which are in fact capable of being executed.

WSIG allows an agent to publish a service description as a web service. The agent sends his service description to DF to save it in Web Services Description Language (WSDL) files [21]. The following figure shows the process for registering the DF of the Interface agent in the WSDL.

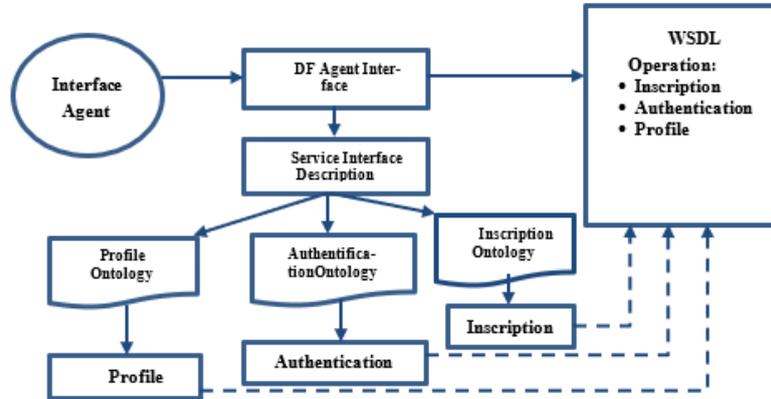


Fig. 13. Example of process for registering the DF of the Interface agent in the WSDL

To save the DF of the Interface agent in WSDL files, the following lines of code must be mentioned:

```

try
{
DFService.register(this,dfadv);
}
catch(Exception e)
{
log.error("Problème pendant Enregistrement Agent Inter-
face" ) ;
doDelete();
}
    
```

Fig. 14. Code of registration of the Interface agent in the WSDL

c) Liaison of agents to their services

WSIG ensure the consistency of the system by linking each agent to its corresponding service through the creation of WSDL files for each service description saved in the DF, and possibly the publication of the services exposed in the UDDI directory.

d) Communication between agents

The JADE platform adopts the communication language FIPA-ACL (Agent Communication Language) [22]. Generally, the communication between agents is done through the ACLMessage class, using the send () method to send a message and the receive() method of the Agent class to receive a message sent by another agent.

The JADE platform provides graphical tools to monitor and control the exchange of messages between the different agents, we use the Sniffer tool which has the role of displaying the interaction flow between the agents and the content of each message exchanged.

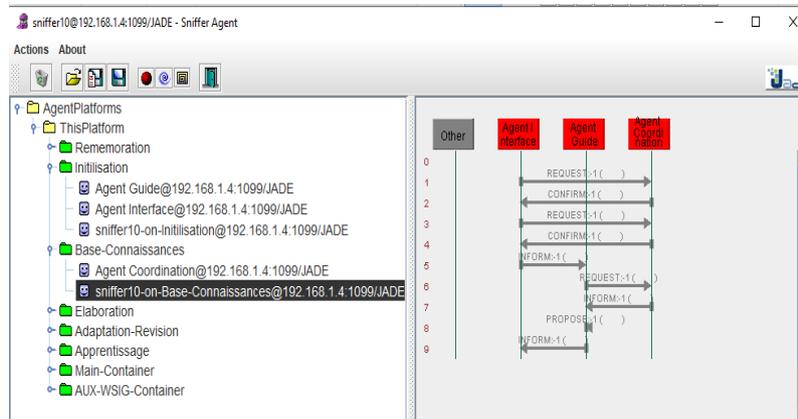


Fig. 15. Communication between the agents of the container Initialization and the Coordination agent

5 Conclusion & Perspectives

Personalized monitoring in real time of the learner in an adaptive learning system allows to offer him an adapted learning to his profile and needs. We have proposed an architecture of a multi-agent adaptive learning system based on IHCBR which takes into account the dynamic change of the behaviors of learners and the other successful experiences of learners. The Retrieve step of IHCBR uses the K-NN algorithm to facilitate the filtering of the learners nearest to the learning situation with similar behaviors, these learners are grouped in homogeneous clusters using the FCM method. Our system contains a set of agents who communicate through an incremental hybrid cycle, using a classic and dynamic cycle of CBR in order to ensure synchronization and coordination between the tasks performed in each step of IHCBR.

We used the JADE platform to develop our multi-agent adaptive learning system and the WSIG component to ensure interaction and communication between JADE agents and the web by Viewing the tasks / activities performed by agents as web services in order to properly exploit the characteristics, capacities and skills of agents.

Our future work consists in testing the effectiveness of our multi-agent adaptive learning system on real cases of learners in order to validate our approach in a concrete way as well as to patent it in the form of a generic and adaptive prototype by using other types of agents in the latter, such as mobile agents, to make our application more general.

6 References

- [1] FORD, Nigel CHEN, Sherry Y. Individual differences, hypermedia navigation, and learning: an empirical study. *Journal of educational multimedia and hypermedia*, 2000, vol. 9, no 4, p. 281-311.
- [2] Blazheska-Tabakovska, Natasha, Ivanovic, Mirjana, Klasnja-Milicevic, Aleksandra, et al. Comparison of E-Learning Personalization Systems: Protus and PLemSys. *International*

- Journal of Emerging Technologies in Learning (iJET), 2017, vol. 12, no 1, p. 57-70. <https://doi.org/10.3991/ijet.v12i01.6085>
- [3] Kolodner J. L. (1993), Case-Based Reasoning. Morgan Kaufmann.
- [4] Huget, Marc-Philippe, James Odell, Bruce R. Poulton, Marian Nodine, Stephen Cranefield, Ronit Ben-Bassat Levy and Lin Padgham. «Fipa modeling: Interaction diagrams. » (2003).
- [5] Board. J, Jade web services integration gateway (wsig) Guide. 2008 Telecom Italia, Last update: 12-Mar-2013.
- [6] Tari, Robert Meersman Zahir, Herrero, Pilar, et al. On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. 2007.
- [7] Graham. S, Davis. D, Simeonov .S, Daniels .G, Brittenham .P, Nakamura .Y, Fremantle .P, König .D, Zentner .C, Building Web Services with Java MAKING SENSE OF XML, SOAP, WSDL, AND UDDI, Sams Publishing, 2005.
- [8] Gudgin .M, Hadley .M, Mendelsohn.N., Moreau .J, Frystyk Nielsen .H, Karmarkar .A, Lafon .Y, SOAP Version 1.2 Part 1 : Messaging Framework (Second Edition), <https://www.w3.org/TR/2007/REC-soap12-part1-20070427/>, 2007.
- [9] Chinnici .R, Gudgin .M, Moreau .J, Schlimmer .J, Weerawarana .S, Web Services Description Language (WSDL) Version 2.0 Part 1 : Core Language, <https://www.w3.org/TR/2003/WD-wsdl20-20031110/>, 2003
- [10] Robert H. Guttman, Alexandros G. Moukas, and Pattie Maes: Agent-mediated Electronic Commerce: A Survey, Software Agents Group, MIT Media Laboratory, (1998). <https://doi.org/10.1080/10864415.2000.11518369>
- [11] MOSTEFAOUI S. K., MOSTEFAOUI G. K., « Towards A Contextualisation of Service Discovery and Composition for Pervasive Environments », Workshop on Web services And Agent-based engineering, Melbourne, Australia, 2003.
- [12] JIN T., GOSCHNICK S., « Utilizing Web Services in an Agent Based Transaction Model (ABT) », Workshop on Web services And Agent-based engineering, Melbourne, Australia, 2003. https://doi.org/10.1007/0-387-23344-x_13
- [13] PETRONE G., « Managing flexible interaction with Web Services », Workshop on Web services And Agent-based engineering, Melbourne, Australia, 2003.
- [14] LYELL M., ROSEN L., Casagni-Simkins M., Norris D., « On Software Agents and Web Services: Usage and Design Concepts and Issues », Workshop on Web services And Agent-based engineering, Melbourne, Australia, 2003.
- [15] Felder, R.M. and L.K. Silverman, Learning Styles and Teaching Styles in Engineering Education, Presented at the 1987 Annual Meeting of the American Institute of Chemical Engineers, New York, Nov. 1987.
- [16] Tsortanidou, Xanthippi, Karagiannidis, Charalampos, et Koumpis, Adamantios. Adaptive educational hypermedia systems based on learning styles: The case of adaptation rules. International Journal of Emerging Technologies in Learning (iJET), 2017, vol. 12, no 05, p. 150-168. <https://doi.org/10.3991/ijet.v12i05.6967>
- [17] Bezdek, James C. Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media, 2013.
- [18] Beloufa, Fayssal. Conception d'un classifieur à base des règles floues. 2016. Thèse de doctorat. 19/01/2017.
- [19] Nihad, El Ghouch, Mohamed, Kouissi, et El Mokhtar, En-Naimi. Designing and modeling of a multi-agent adaptive learning system (MAALS) using incremental hybrid case-based reasoning (IHCBR). International Journal of Electrical & Computer Engineering (2088-8708), 2020, vol. 10. <https://doi.org/10.11591/ijece.v10i2.pp1980-1992>
- [20] Bahri, Mohamed Rédha. Une approche intégrée Mobile-UML/Réseaux de Pétri pour l'analyse des systèmes distribués à base d'agents mobiles 2010.

- [21] Chinnici. R, Gudgin .M, Moreau .J, Schlimmer .J, Weerawarana .S, Web Services Description Language (WSDL) Version 2.0 Part 1 : Core Language, 2003, <https://www.w3.org/TR/2003/WD-wsdl20-20031110/>.
- [22] Bellifemine, Fabio, Poggi, Agostino, et Rimassa, Giovanni. JADE–A FIPA-compliant agent framework. In: Proceedings of PAAM. 1999. p. 33.

7 Authors

El Ghouch Nihad is a PhD student in LIST (Laboratoire d'Informatique, Systèmes et Télécommunications) Laboratory, Department of Computer Science. Faculty of Sciences and Techniques, Tangier, Morocco, at the University Abdelmalek Essaadi. The research topics of interest are adaptive learning, personalized learning.

En-Naimi El Mokhtar is a Professor in UAE, Faculty of Sciences and Technologies of Tangier, Department of Computer Science, since 2000 (Temporary: from 2000 to 2003 and Professor Permanent: since 2003/2004 until Now). He is a Head of Computer Science Department, since October 2016 until now. He was responsible for a Bachelor of Science and Technology, BST Computer Engineering (“License LST-GI”), from January 2012 to October 2016. He is also a founding member of the Laboratory LIST (Laboratoire d'Informatique, Systèmes et Télécommunications), the University of Abdelmalek Essaâdi, FST of Tangier, Morocco. He is also an Expert Evaluator with the ANEAQ, since the academic year 2016/2017 until now, that an Expert of the Private Establishments belonging to the territory of the UAE and also an Expert of the Initial or Fundamental Formations and Formations Continuous at the Ministry of Higher Education, Scientific Research and Executive Training and also at the UAE University and the FST Tangier since 2012/2013 until Now. He is an Author/Co-Author of several Articles, published in The International Journals in Computer Science, in particular, in MAS, Cases Based Reasoning (CBR), eLearning, MOOC, Big DATA, Data-mining, Wireless Sensor Network, VANet, MANet, Smart City, etc. He is also Director of several Doctoral Theses in Computer Science. In addition, he is an associate member of the ISCN - Institute of Complex Systems in Normandy, the University of the Havre, France, since 2009 until Now. Email address: en-naimi@uae.ac.ma

Kouissi Mohamed is a PhD student in LIST (Laboratoire d'Informatique, Systèmes et Télécommunications) Laboratory, Department of Computer Science. Faculty of Sciences and Techniques, Tangier, Morocco, at the University of Abdelmalek Essaadi. He had State Engineer Diploma, Specialized in Computer Science, laureate of Tangiers National School of Applied Sciences in 2011. He had also, 8 years of experience in developing mobile applications for smartphone / tablet / iPhone / iPad with Android/iOSTechnologies. In addition, he is an Android Lead project in a French company "LeKiosk" and a Developer of Android / iOS mobile applications for Presstalis (Leader of press distribution in France). The research topics of interest are Multi-Agent Systems (MAS), Case Based Reasoning (CBR), eLearning...etc. Email address: mohamed.kouissi@gmail.com

Article submitted 2020-02-19. Resubmitted 2020-03-25. Final acceptance 2020-03-30. Final version published as submitted by the authors.