ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 7, January 2017

Multi Agent Systems Applications

Sophia Faris, Hicham Medromi, Adil Sayouti
Laboratoire de recherche en ingénieurie, ENSEM,
Université Hassan II, Casablanca, Morocco

Abstract*— The multi-agent systems approach of knowledge level co-operation between autonomous agents promises significant benefits to the complex systems such as enhanced interoperability, scalability and reconfigurability. Nowadays, large intelligent systems are very complex. They process a huge amount of information and make very heavy calculations. Recently, new technologies in artificial intelligence, such as Agents can be useful to model these systems and allow them to gather the prior knowledge they need. Agents are autonomous, pro-active, communicative, goal-directed, flexible, capable of learning, and mobile. This paper discusses a formal approach of agent-based modeling for intelligent systems, which describes design level precautions, challenges and techniques using autonomous agents, as its fundamental modeling abstraction. We describe different architectures, based on multi-agents systems, proposed by the system architecture team of ENSEM, Hassan II University, in different domains: Governance, Risk and Compliance of Information Systems, distance learning and the remote control.

***Index Terms*—Information systems, information security risk management, multi agent systems.**

I. INTRODUCTION

In the recent years, agent oriented programming is one of the most important areas of research and development that have emerged in information technology.

A. Approach agent vs. approach object

The multi-agents system is considered as an object-oriented system that is associated to an intelligent meta-system.

By this way, an agent is viewed as an object that has a layer of intelligence, comprising a number of capabilities such as uniform communication protocol, perception, reaction and deliberation, all of them not inherent to objects. However, the Agent oriented approach (AOP) has code, states and agent invocations.

The agents also have individual rules and goals to make them appear like active objects within initiative. In AOP the class is replaced by role, state variable with belief/knowledge and method with message.

The role definitions describe the agent capability and the information needed to desired results. In order to the agents act with intelligence in their environment, the idea is to develop the complex entities and provide the agents with the knowledge and beliefs to be able to achieve their desires. The table 1 illustrates the differences between the agent approach and the object approach.

Criteria	Agent approach	Object approach
Basic Unit	Agent	Object
Unit state	Mental components	Unconstrained
Communication paradigm	Peer-peer	Client-server
Communication mode	Message passing	Message passing
Communication type	Local (mobile) + remote (static)	Mostly remote
API	Uniform method call	Unconstrained
Method constraints	Honesty, consistency, etc.	None
Message type	Speech Acts (ACL)	Unconstrained
Mobility	Autonomy and mobility-related metadata	No autonomy or mobility related meta data
Inheritance	Mental states	Methods and attributes
Intelligence	Intelligent operations	Not always present

Table 1. Agent approach vs. Object approach

II. AGENT AND MULTI AGENT SYSTEMS

Jennings and Wooldridge [Jennings & Wooldridge 1998] have defined an agent as "a computer system located in certain environment which is able to act autonomously in this environment, in order to meet its design goals". Agents have the following main properties and characteristics:

- **Autonomy** : agents encapsulate a state (which is not available to other agents), and make decisions on what to do based on this state, without direct human intervention or other persons;
- **Social ability**: agents interact with other agents (and possibly humans) via some kind of agent communication Language, and generally have the opportunity to participate in social activities (such as cooperation for solving problems or negotiating) to achieve their goals.
- **Reactivity**: agents are put in an environment (which may be the physical world, a user via a graphical interface, a collection of other agents, the internet, or perhaps many of these combinations), are able to perceive this environment (through the use of potentially imperfect sensors), and are able to respond to timely changes that occur in it.
- **Proactivity** : Agents do not simply act in response to their environment; they are able to solve a problem by taking the initiative.

A multi-agent system (MAS) is a system composed of several intelligent agents that interact with each other. They can be used to solve problems that are difficult or impossible to solve for an individual agent or monolithic system. Multi-agent systems are open and scalable systems that enable the implementation of autonomous and proactive software components. They are characterized by the local autonomy, social interaction, adaptability, robustness and scalability, and for these reasons, they are a very promising paradigm to address the challenges facing automation and check systems. A multi agent system are ideally suited to represent and solve problems that have multiple perspectives.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 7, January 2017

A. *Kinds of agents' architectures*

- **Reactive architecture**: Reactive agent architecture is based on the direct mapping of situation to action.

Agent responses to changes in the environment in a stimulus-response based.

These agents have no representation of their environment, no knowledge base and no memory.

Brook's subsumption architecture is known as the best pure reactive architecture (Brooks, 1986).

This architecture was developed by Brook who has critiqued on many of the drawbacks in logic-based architecture.

Figure 1 illustrates an example of reactive architecture. The figure shows that each of the percept situation is mapped into an action which specifically responses to it.

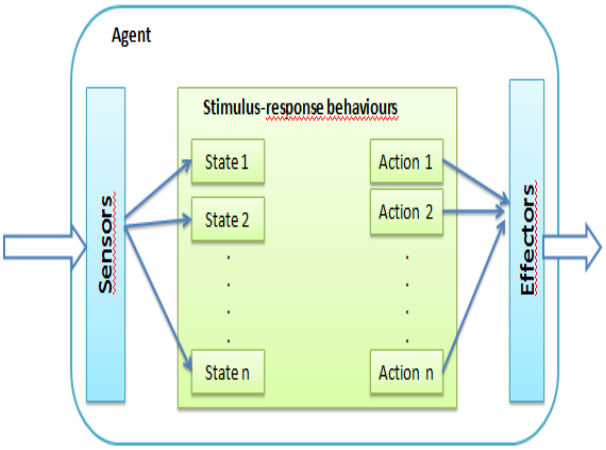

Fig 1: Reactive agent architecture

- **Deliberative Reasoning architectures**:

This architecture is based on the traditional artificial symbolic approach by representing and modeling the environment and the agent behavior with symbolic representation.

Thus, the agent behavior is based on the manipulation of the symbolic representation.

The internal process of the deliberative agent is more complex than the reactive agent.

The difference lies in fact that the deliberative agent possesses internal image of the environment and is capable to plan its actions.

The most commonly used architecture for implementing such behavior is Belief, Desire, and Intention Software model (BDI).

Agents' beliefs about the world, desires and intentions are internally represented and practical reasoning is applied to decide, which action to select.

According to Wooldridge's definition, a deliberative agent is "one that possesses an explicitly represented, symbolic model of the world, and in which decisions (for example about what actions to perform) are made via symbolic reasoning".

Figure 2 illustrates an example of deliberative agent.

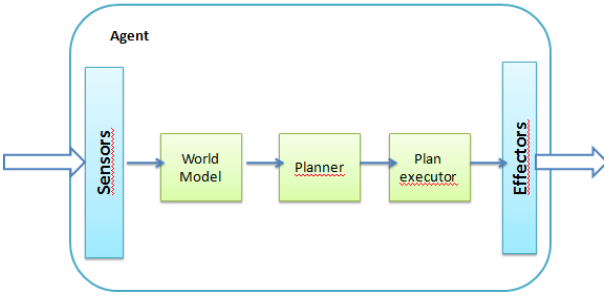

Fig 2: Deliberative agent architecture

- **Hybrid architectures**: Hybrid architecture is an agent architecture which allows both reactive and deliberate agent behavior. It combines both the advantages of reactive and logic-based architecture and at the same time attenuates the problems in both architectures.

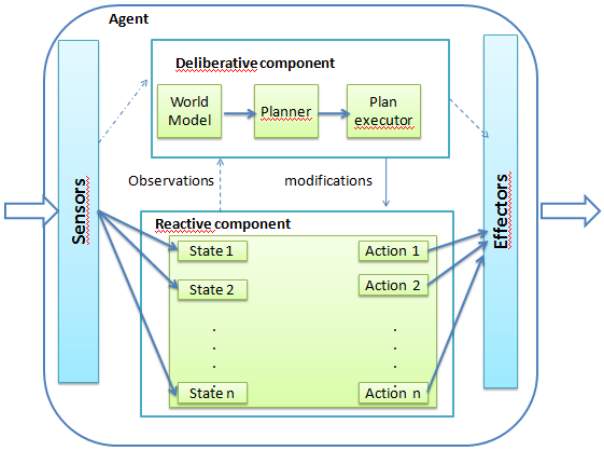

Fig 3: Hybrid agent architecture

- Cognitive architectures: Cognitive architectures are used to construct intelligent system/agent that models human performance (Newell, 1990, Meyer & Kieras, 1997).

It refers to a theory about the structure of the human mind.

Its main goal is to summarize the various results of cognitive psychology in a comprehensive computer model.

Successful cognitive architectures include Adaptive Control Thought (ACT).

The construction of intelligent agent using cognitive architecture focuses on the cognition part, the simulation and modeling of human behavior.

B. *Agent communication*

The cooperation in a multi-agents system is based on the communication and the interaction between agents.

The agent communication, also known as the agent-based messaging paradigm [6], provides a universal messaging language with a consistent speech-act-based, uniform messaging interface for exchanging information, statically or dynamically, among software entities. Agent communication has the following advantages over the traditional client-server (RPC) based communication:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 7, January 2017

- De-centralized, peer-peer communication, as opposed to the traditional client-server roles
- Asynchronous exchange of messages
- Universal message-based language with speech-act-based interface
- Single method invocation for all types of message exchanges (FIPA : Foundation for Intelligent Physical Agents)

The communication involves at least two parties: a sending agent that generates the information and transmits it and a receiving agent that receives the message and uses the information.

The information that is exchanged between the Communicating parties may be formally coded into a Universally understood agent communication language (ACL) with a speech act based interface. The sending agent on generating this ACL coded message string invokes the message method of the recipient and passes the string through it (FIPA framework). The receiving agent, on receiving this message, decodes the information and then performs the necessary actions. In case of a bidirectional communication, it may communicate the result back to the sender by reciprocating the same process.

C. Agent communication language

Agent communication, under this paradigm, is accomplished through the use of three components: ontology, content language, and agent communication language. Ontology enumerates the terms comprised by the application domain.

The content language is used to combine terms in the Ontology into meaningful sentences in the language as defined by the grammar. Sometimes the two are so tightly coupled that they become one. Finally, the agent communication language acts as a medium for exchanging dialogs among agents, containing sentences of the content language. It provides the outer encoding layer, which determines the type of agent interaction, identifies the network protocol with which to deliver the message, and supplies a speech act also known as communicative act or per formative. The communicative act indicates whether the message is an assertion, a query, a command or any other acceptable speech form. ACLs range from some form of primitive communication to elaborated standards. Two of the most widely used ACLs are knowledge query manipulation language (KQML) and FIPA ACL [7]. Knowledge interchange format (KIF) is often used as a content language with KQML. Likewise, semantic language (SL) is often used to represent the application domain, even though the FIPA ACL specification document does not make any commitment to a particular content language.

D. Agent coordination

Coordination is an important aspect of multi-agent systems.

MAS provide appropriate concepts for realizing systems that offer inherently non-functional requirements such as scalability, robustness and failure tolerance.

However, agents' activities need to be coordinated, in order to ensure the functionality of MAS on a global level. Coordination is therefore one of the key aspects of MAS and can be defined as the management of dependencies [Malone and Crowston, 1994]. Agents' coordination is about managing their activities. The type of coordination is differentiated by the fact if it is realized by direct or indirect interaction. In recent years, researchers have given importance to distributed coordination of MAS, because of its important applications in many areas. The objective of distributed coordination of MAs is allowing agents to cooperatively fulfill a task in a distributed manner regarding a certain quantity of interest that depends on the states of all agents.

E. Multi agents platforms overview

Agents-based systems are often difficult to implement directly with standard programming languages such as Java or C ++.

Several development platforms (software elements offering services for the development of MAS) of different types are developed recently for agent oriented programming.

This section presents some of the platforms that are available today.

- ✓ **MADKIT**: The multi agent Development Kit is an open source modular and scalable multi agent platform written in Java and built upon the AGR (Agent/Group/Role) organizational model.
  MaDKit agents play roles in groups and thus create artificial societies. It provides general agent facilities, such as lifecycle management, message passing and distribution, and allows high heterogeneity in agent architectures and communication languages, and various customizations.
- ✓ **JADE**: is a framework fully implemented in Java.

It simplifies the implementation of multi-agent systems through a middle-ware that claims to comply with the FIPA specifications. The agent platform can be distributed across machines (which do not even need to share the same OS) and the configuration can be controlled via a remote GUI.

JADE is industry-driven and currently it is the most popular FIPA-compliant agent platform in academic and industrial community.

The tool has three main modules (required by FIPA standards). The DF "Director Facilitor" provides a yellow pages service to the platform. The "Agent Communication Channel" ACC manages communication between agents. The AMS "Agent Management System" monitors the registration of agents, their authentication, access and use of the system. The agents communicate through the FIPA ACL language.

- ✓ **Agent Builder**: AgentBuilder is a complete development environment [AgentBuilder, 2000]. It is another case of traditional agent platform that can be used in numerous simulation cases.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 7, January 2017

✓ It is a KQML-compliant integrated tool suite for constructing intelligent software agents, allowing software developers with no background in intelligent systems or intelligent agent technologies to quickly and easily build intelligent agent-based applications.

✓ The behavior of agents is developed using the BDI model and the AGENT-0 language.

✓ KQML is used as a communication language between agents. KQML is a language and protocol for communication among software agents and knowledge-based systems (Finin et al. 1994). AgentBuilder is a complex tool that requires significant learning efforts and good knowledge in the field of multi-agent systems for efficient use. It is limited to scalability, deployment and reusability.

✓ **Zeus**: Zeus is a complete environment [Collis, 1999] which uses a methodology called "Role Modeling" for the development of collaborative systems.

✓ The agents have three layers. The first layer is that of the definition where the agent is seen as an autonomous entity capable of reasoning in terms of his beliefs, resources and preferences.

✓ The second layer is that of organization. In this, we decide the modes of communication between agents, protocols, coordination and other mechanisms of interaction. The third layer is the coordination layer, each agent is considered as a social entity, ie in terms of its negotiation and coordination skills.

The tool is one of the most complete. The various stages of development are carried out within several editors: ontology, description of tasks, organization, definition of agents, coordination, facts and variables as well as constraints. However, the development of MAS with Zeus is conditional on the use of the "Role Modeling" approach. The tool is quite complex and its control requires a lot of time.

III. THE PROPOSED ARCHITECTURES BASED ON MULTI AGENT SYSTEMS

In this section, we describe the architectures proposed by the system architecture team in ENSEM (Hassan II University).

In this section, we are interested to the following domains: Governance Risk Compliance, Distance education, and remote control.

A. MAS application in the Governance, Risk and Compliance of the information systems

GRC (Governance, Risk and Compliance) is an integrated and holistic approach to the organization that ensuring that the organization is ethically correct and consistent with its risk desire, internal policies and external regulations by aligning strategy, processes, technology, and people, improving efficiency and effectiveness [1].

The objective of this work is to propose an integrated IT GRC architecture for a high level IT GRC management. This solution is based on existing frameworks for the separate topics of IT governance, IT risk and IT compliance.

Nowadays, information systems are everywhere. Information plays a vital aspect in every organization

In the current context of globalization characterized by continuous evolution, harsh competition and uncontrolled external and internal environments, information systems are required to maintain a satisfactory level of performance in order to contribute to the continuity of organizations' business activities.

New technologies in artificial intelligence such as multi agent systems try to solve information security management issues. Indeed, MAS can be useful for designing efficiently and maintaining secure information systems.

Agents are autonomous, proactive, communicative, goal-directed, flexible, capable of learning, and mobile. All these features allow governing and managing risks within information systems in an effective way.

The interface agent is the high level of our architecture, it is responsible of setting the user's goal (Governance, risk or compliance).

The agent manager is linked to the knowledge base which contains a set of standards in IT GRC. This agent is responsible of communicating with the three MAS (IT G, IT R, IT C) based on the user's objective. MAS-EAS IT GRC is composed of three multi agent systems : MAS-IT G, MAS-IT R, MAS IT C.

✓ **MAS-IT G**: is composed of two agents:

Agent dashboard: It is responsible for updating the data about performance indicators.

Agent report: It is responsible for making reports concerning which methodology the user choose to implement.

✓ **MAS-IT R**: is composed of two agents:

Agent risks: It is responsible for managing the risks in the MAS-IT GRC by following the steps: identifying the assets of the organizations and evaluating them; identifying the vulnerabilities and evaluating them; identifying the threats that exploit the vulnerabilities and evaluating them; and finally identifying the risks that impact the assets.

After collecting these data, this agent is responsible to send the identified risks and their evaluation to the agent controls.

Agent controls: It is responsible for mitigating the risks, by associating the relevant control to the risk identified earlier.

✓ **MAS-IT C**: is composed of one agent:

Agent compliance: It is responsible of giving the user a set of questionnaire about the law or the standard he choose.

After that, based on the answers of the user, the system gives the level of compliance to the standard or the law chosen before.

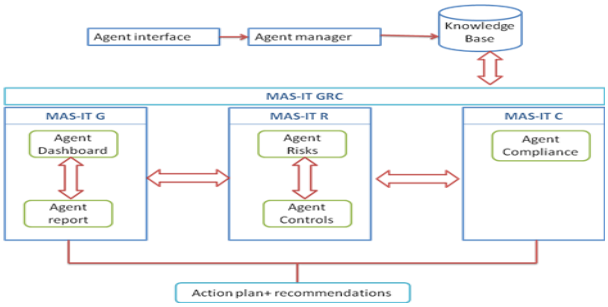

Fig 4: MAS-EAS IT GRC architecture

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 7, January 2017

B. MAS application to distance education

The objective of this work is to create an innovative educational tool to allow students to perform remote laboratory experiments on autonomous and teleoperated mobile systems. The solution proposed in this work consists in equipping the mobile systems with a high degree of local intelligence in order for them to autonomously handle the uncertainty in the real world and also the arbitrary networks delay.

The remote laboratories provide a live performance laboratory accessible via Internet, which can be used to cover the experimental issues in any tele-education system.

Clearly as bandwidth increases and higher speed network access reaches users; these factors play an important role in user adoption of remote laboratories. The concept of remote laboratories is defined as a mechatronic workspace for distance collaboration and experimentation in research or other creative activity, to generate and deliver results using distributed information and communication technologies. To implement a remote laboratory, a common Internet-based teleoperation model is used as shown in the figure below.[6]

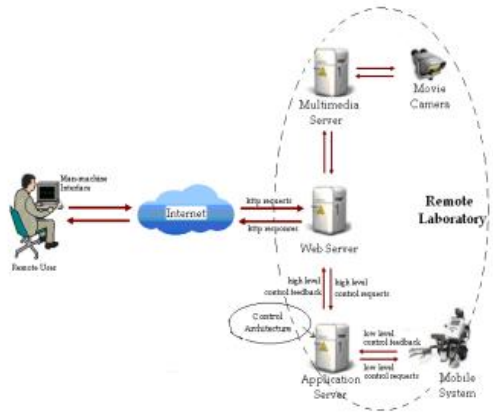

Fig 5: System architecture [6]

The Remote user, through his Internet navigator, addresses a http request to a Web server and downloads an application on his work station. A connection is then established towards the server in charge of the management of the mobile system to control. The user is then able to take the remote control of it. In parallel, other connections are also established towards multi-media servers broadcasting signals (video, sound) of the system to be controlled.

The Internet network (network without quality of service) limits the quantity of information that can be transmitted (bandwidth) and introduces delays which can make the remote control difficult or impossible. The solution proposed, through this work, to face the limitations of the Internet, is founded on the autonomy and the intelligence, based on multi-agents systems, granted to the mobile system in order to interact with its environment and to collaborate with the remote user. The need that consists in wanting to assign to the mobile system the maximum of autonomy and intelligence brought us to examine in the detail the choice of a remote control architecture [4].

C. MAS application to remote control

The objective of this work is to present a hybrid architecture called EAAS for EAS Architecture for Autonomous system, including a deliberative part (Actions Selection Agent) and a reactive part.

The deliberative part which uses methods of artificial intelligence contains a path planner, a navigator and a pilot. The reactive part is based on direct link between the sensors (Perception Agent) and the effectors (Action Agent).

Fundamental capacities of our architecture encompass autonomy, intelligence, modularity, encapsulation, scability and parallel execution.[4]

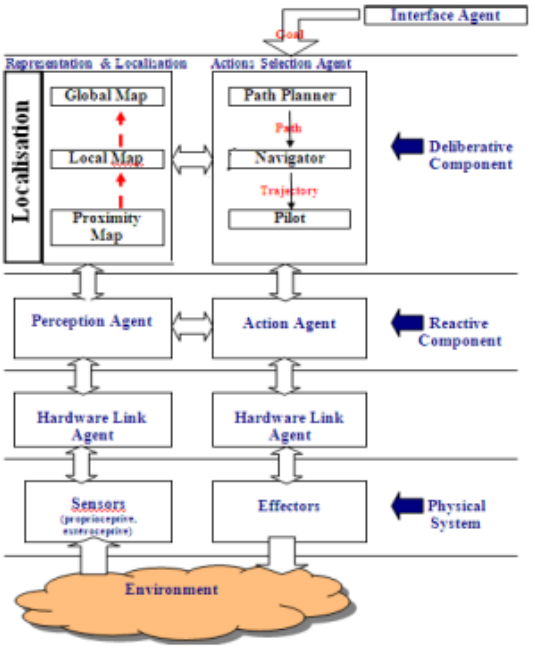

Fig 6: EAAS architecture [4]

EAAS architecture consists in five agents: interface agent, actions selection agent, perception agent, action agent and hardware link agent. The interface agent is the high level of our control architecture. It must generate a succession of goal, or missions for the actions selection agent, according to the general mission of the robot. It is the "ultimate" robot autonomy concept: the robot generates itself its own attitudes and its own actions by using its own decisions. The perception agent manages the processing of incoming data (the sensor measurements) and creates representations of the environment. The actions selection agent must choose the robot behavior according to all information available and necessary to this choice: the fixed goal, representations and the robot localization. The actions selection agent contains a path planner, a navigator and a pilot. The path planner may take a goal as input and give a path for achieving the goal as output. The Navigator must translate a path into a trajectory for the pilot. The path does not take into account physical constraints of the robot, but the trajectory that it delivers must

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 7, January 2017

Integrate them. The function of the pilot is to convert this trajectory into orders to be performed by the action agent [4].

IV. CONCLUSION

In this paper, we have shown that applications of multi agent systems are various.

MAS can be used to design and develop distributed, autonomous and intelligent architectures in a lot of domains like IT GRC, distance education and remote control.

As a perspective, we intend to increase the intelligence of the agents in our architecture of IT GRC, in order to give the organizations an overview of the threats, vulnerabilities and risks occurring in their information systems.

We will consider applying several paradigms of artificial intelligence such as expert systems to meet the objectives of IT GRC.

ACKNOWLEDGMENT

I would like to thank my advisors Ms. H. Medromi and Ms. A. Sayouti for the efforts they made to review this paper and for the many useful suggestions they gave me during the work on this paper. I would also express my gratitude to every person that has contributed to the achievement of this works.

REFERENCES

[1] Racz, N., Panitz, J.C., Amberg, M., Weippl, E. &Seufert, A. (2010): Governance, Risk & Compliance (GRC) Status Quo and Software Use: Results from a survey among large enterprises. In: ACIS 2010 Proceedings, Paper 21. Retrieved 13 December 2010 from: http://aisel.aisnet.org/acis2010/21

[2] M. Wooldridge. Agents and software engineering. In *AI*IA* Notizie XI(3), 1998 ,pages 31-37.

[3] J. Ferber, (Les systèmes multi-agents, vers une intelligence collective) InterEditions, 1995, pp. 63-144.

[4] A.Sayouti & H. Medromi, *ntechopen,2011*,(Autonomous and Intelligent Mobile Systems based on Multi-agent, Book Chapter in the book, Multi-agent Systems – Modeling Control , Programming, Simulations and Applications) ,.

[5] S.Faris,H. Iguer, H.Medromi and S.Sayouti, New model multi-agent systems based for the security of information system, Proc. IC2INT'13, 2013.

[6] Adil SAYOUTI, Hicham MEDROMI, Sophia FARIS, "Multi-agents Systems : Application to Distance Education", ICEER'13

[7] S.Faris, H.Iguer, H.Medromi and A.Sayouti, Conception d'une Plateforme de gestion des risques basée sur les systèmes multi-agents et ISO 27005, JDTIC, 2013.

[8] S.Faris, H.Medromi and A.Sayouti, Modélisation d'une plateforme (SIGRCI) à base des systèmes multi-agents & ITIL, JDTIC, 2012.

[9] W. Boehmer, Appraisal of the effectiveness and efficiency of an Information Security Management System based on ISO 27001, Proc. Second Int. Conf. Emerging Security Information, Sys. & Technologies. pp: 224-231, 2008.

[10] N.Mayer, P.Heymans, and R.Matulevicius. Design of a modelling Language for Information System Security Risk Management. In Proceedings of the 1st International Conference on Research Challenges in Information Science (RCIS 2007), pages 121-131-,2007.

[11] Hayzelden, A. L.; Bigham J. Software agents for future communication systems. 1st ed. New York: Springer, 1999. Pp. 101

[12] Wooldridge, M. "Conceptualizing and Developing Agents". In Proceedings of the UNICOM Seminar on Agent Software. 1st ed. London, 1995. Pp. 42

[13] Kalliopi Kravari & Nick Bassiliades, "A Survey of Agent Platforms", Journal of Artificial Societies and Social Simulation (JASSS), 2015.

AUTHOR BIOGRAPHY

Sophia Faris is a PhD student in the National Superior School of Electricity and Mechanics (ENSEM), Hassan II University, Casablanca, Morocco. She is an engineer in computer sciences in ENSEM since 2011. In 2013, she got certificates of ITIL V3 and ISO 27002 foundation. Her actual main research interest concern Information Security Risk Management and Security Auditing in Information Systems.

Hicham Medromi received the PhD in engineering science from the Sophia Antipolis University in 1996, Nice, France. He is responsible of the system architecture team of the ENSEM Hassan II University, Casablanca, Morocco. His actual main research interest concern Control Architecture of Mobile Systems Based on Multi Agents Systems. Since 2003 he is a full professor for automatic product and computer sciences at the ENSEM, Hassan II University, Casablanca..

Adil Sayouti received the PhD in computer science from the ENSEM, Hassan II University in July 2009, and Casablanca, Morocco. In 2003 he obtained the Microsoft Certified Systems Engineer (MCSE). In 2005 he joined the system architecture team of the ENSEM, Casablanca, Morocco. His actual main research interests concern Remote Control over Internet Based on Multi agents Systems.

.