AMADE: Developing a Multi-Agent Architecture for Home Care Environments

Juan A. Fraile[1], Javier Bajo[1], Juan M. Corchado[2]

[1]Pontifical University of Salamanca, Compañía 5, 37002 Salamanca, España
[2]Departamento de Informática y Automática, Universidad de Salamanca,
Plaza de la Merced s/n, 37008 Salamanca, España
{jafraileni, jbajope}@upsa.es, corchado@usal.es

Abstract. Obtaining novel and effective solutions for home care is one of the aims of the Ambient Intelligence. Elderly and disabled people require special care services, and the emerging technologies facilitate the development of intelligent environments at their homes. This paper presents AMADE, a multi-agent architecture for the control and supervision of home care environments. AMADE integrates an innovative alert management system as well as automated identification, location, and movement control systems. In this sense, this work proposes a new perspective to tackle the problem of home care. The architecture has been successfully tested in a real environment and the results obtained are presented within this paper.

Keywords: Dependent environments, Ambient Intelligence, Multiagent Systems, Home Care

1 Introduction.

There is an ever growing need to supply constant care and support to the disabled and elderly and the drive to find more effective ways to provide such care has become a major challenge for the scientific community [5]. In recent years, there has been an important growth in the field of Ambient Intelligence (AmI) [1] [13], involving major changes in the daily lives of people. The vision of Ambient Intelligence implies the creation of intelligent spaces where users interact in a natural way with computational systems and communication technologies, which become invisible and ubiquitous. The technology is adapted to individuals and their context, acting autonomously, and facilitating their daily tasks. One of the main aims of Ambient Intelligence focuses in building systems that support the activities of daily living in an efficient way. For example, activities related to the home automation [10].

The complexity of these systems requires advanced control architectures, which have gained increasing importance in recent years. These architectures must provide novel structures, information exchange mechanisms and computer resources management. Besides, most of the current systems require concurrent work to provide real time solutions. This paper presents Architecture Multi-Agent Dependent Environments (AMADE), a novel architecture specifically designed to be implemented in Ambient Intelligence home care environments. AMADE integrates an alert management system and an identification, location and movement control mechanism based on a multi-agent system. These mechanisms facilitate most of the common tasks required in home care environments, as patients monitoring and tracking, as well as quick response to problematic situations. Multi-agent systems are very appropriate to satisfy these needs, due to their characteristics [15]. One of the main contributions of AMADE is the use of both reactive and deliberative agents, which facilitates advanced reasoning capabilities together with real-time reactive behaviors. These are two important characteristics that must be taken into account in the development of intelligent environments.

The paper is organized as follows: next section presents the problem that motivates this work. The third section describes the proposed architecture and the fourth section describes the location, identification and alert systems integrated in the architecture. The fifth section presents as the architectures applies to real case. Finally the last section presents the results and conclusions obtained after applying the proposed architecture to a real dependent environment.

2 General description of the problem.

The use of intelligent agents is an essential component for analyzing information on distributed sensors [15]. These agents must be capable of both independent reasoning and joint analysis of complex situations in order to be able to achieve a high level of interaction with humans [3]. Although multi-agent systems already exist and are capable of gathering information within a given environment in order to provide medical care [7], there is still much work to be done. It is necessary to continue developing systems and technology that focus on the improvement of services in general. After the development of the internet there has been continual progress in new wireless communication networks and mobile devices such as mobile telephones and PDAs. This technology can help to construct more efficient distributed systems capable of addressing new problems [8].

Hybrid architectures try to combine deliberative and reactive aspects, by combining reactive and deliberative modules [6]. The reactive modules are in charge of processing stimuli that don't need deliberation, whereas the deliberative modules determine which actions to take in order to satisfy the local and cooperative aims of the agents. The aim of modern architectures like Service Oriented Architecture (SOA) is to be able to interact among different systems by distributing resources or services without needing to consider which system they are designed for. An alternative to these architectures are the multi-agent systems, which can help to distribute resources and to reduce the centralization of tasks. Unfortunately the complexity of designing multi-agent architecture is great since there are no tools to either help programme needs or develop agents.

Multi-agent systems combine aspects of both classic and modern architectures. The integration of multi-agent systems with SOA has been recently investigated [2]. Some researchers focus on the communication among these models, whereas others focus on the integration of distributed services, especially web services, in the agents' structure [4] [13]. These works provide a good base for the development of multi-agent systems. Because the majority of them are in the development stage, their full potential in a real environment is not known. AMADE has been implemented in a real environment and not only does it provide communication and integration among distributed agents, services and applications, but it also provides a new method for facilitating the development of multi-agent systems, thus allowing the agents and systems to function as services. AMADE also implements an alert and alarm system across the agent's platform, specially designed to be used by mobile devices. The platform agents manage this service and determine the level of alert at every moment so that they can decide who will receive the alert and when. The alerts are received by the subscribed in the system that have associated such alerts. Hours sending alerts depend on the system parameters and the urgency of the warning. In order to identify each user, AMADE implements a system based on Java Card [16] and RFID (Radio Frequency IDentification) microchip technology in which there will be a series of distributed sensors that provide the necessary services to the user.

3 AMADE Multi-Agent Architecture.

The design of AMADE model architecture takes into account flexibility and extensibility requirements. These requirements are common to all kinds of distributed systems, but especially apply to this application domain, due to its intrinsic characteristics, such as different national and local regulation frameworks, the heterogeneity of health centres and communities involved in care service delivery and the patient's health status development over time.

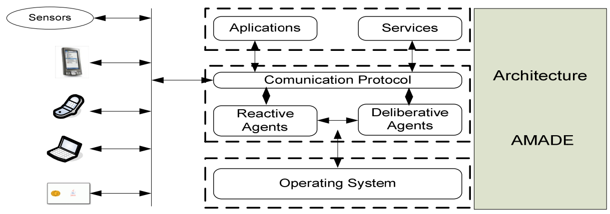

Fig. 1. AMADE Framework.

AMADE incorporates intelligent agents, identification and localization technologies, wireless networks and mobile devices. Additionally, it provides access mechanisms to multi-agent system services, through mobile devices, such as mobiles phones or PDAs. Access is provided via wi-fi wireless networks, a notification and alarm management module based on SMS (Short Message Service) and MMS (Multimedia Messaging System) technologies, and user identification and localization system based on Java Card and RFID technologies. This system is dynamic, flexible, robust and very adaptable to changes in the context. The AMADE architecture is composed of four basic blocks that can be seen in Figure 1: Applications, Services, Agents Platform and Communication Protocol. These blocks constitute the whole functionality of the architecture. The applications represent all programs that can be used to exploit the system functionalities. The services represent the functionalities that the architecture offers. The agent's platform is the core of the architecture and integrates two types of agents, each of which behaves differently for specific tasks, as shown in Figure 2.

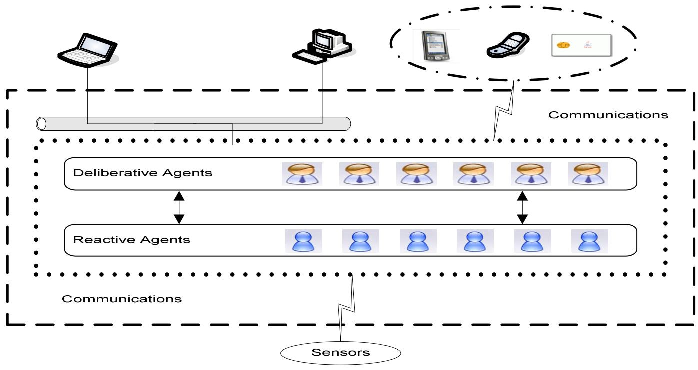

Fig. 2. Agents platform structure in the AMADE architecture.

The first group of agents is made up of deliberative BDI agents, which are in charge of the management and coordination of all system applications and services. These agents are able to modify their behaviour according to the preferences and knowledge acquired in previous experiences, thus making them capable of choosing the best solution. Deliberative agents constantly deal with information and knowledge. Because they can be executed on mobile devices, they are always available and they provide ubiquitous access for the users. There are different kinds of agents in the architecture, each one with specific roles, capabilities and characteristics. This fact facilitates the flexibility of the architecture to incorporate new agents. However, there are pre-defined agents which provide the basic functionalities of the architecture:

- CoAp Agent: This agent is responsible for all communications between applications and the platform. Manages the incoming requests from the applications to be processed by services. It also manages responses from services to applications. CoAp Agent is always on listening mode. Applications send XML messages to the agent requesting for a service.
- CoSe Agent: It is responsible for all communications between services and the platform. The functionalities are similar to CoAp Agent but backwards. This agent is always on listening mode waiting for responses of services. Manager Agent indicates CoSe Agent the service that must be invoked. Then, CoSe Agent sends an XML message to the service.
- Directory Agent. Manages the list of services that can be used by the system. For security reasons, the list of services is static and can only be modified

manually, however services can be added, erased or modified dynamically. The list contains the information of all trusted available services.

- Supervisor Agent. This agent supervises the correct functioning of the agents in the system. Supervisor Agent verifies periodically the status of all agents registered in the architecture by means of sending ping messages. If there is no response, the agent kills the agent and creates another instance of that agent.
- Security Agent. This agent analyzes the structure and syntax of all incoming and outgoing XML messages. If a message is not correct, the Security Agent informs the corresponding agent that the message cannot be delivered.
- Manager Agent. Decides which agent must be called taking into account the users preferences. Users can explicitly invoke a service, or can let the Manager Agent decide which service is better to accomplish the requested task. If there are several services that can resolve the task requested by an application, the agent selects the optimal choice using a strategy based on the analysis of the efficiency of the services. An optimal choice service has higher and better performance that other. Manager Agent has a routing list to manage messages from all applications and services.
- Interface Agent. This kind of agent has been designed to be embedded in users' applications. Interface agents communicate directly with the agents in AMADE so there is no need to employ the communication protocol, but FIPA ACL specification. The requests are sent directly to the Security Agent, which analyzes the requests and sends them to the Manager Agent.

The second group is made up of reactive agents. Most of the research conducted within the field of multi-agent systems focuses on designing architectures that incorporate complicated negotiation schemes as well as high level task resolution, but don't focus on temporal restrictions. In general, the multi-agent architectures assume a reliable channel of communication and, while some establish deadlines for the interaction processes, they don't provide solutions for limiting the time the system may take to react to events. It is possible to define a real-time agent as an agent with temporal restrictions for some of its responsibilities [12]. From this definition, we can define a real-time multi-agent system (Real Time Multi-Agent System, RT-MAS) as a multi-agent system in which at least one of the agents is a real-time agent. The use of RT-MAS makes sense within an environment of critical temporal restrictions, where the system can be controlled by autonomous agents that need to communicate among themselves in order to improve the degree of system task completion.

Communication protocol allows applications, services and sensors to be connected directly to the platform agents. The protocol presented in this work is based on the SOAP standard and allows messages to be exchanged between applications and services. The interaction with environmental sensors requires Real-time Transport Protocol (RTP) [11] [6] which provides transport functions that are adapted for applications that need to transmit real-time data such as audio, video or simulation data, over multicast or unicast network services. The communication~~s~~ between agents within the platforms follows the FIPA ACL (Agent Communication Language) standard. This way, the applications can use the platform to communicate directly with the agents.

4 Improving AMADE with Location, Identification and Alert Systems

The alert system is integrated into the AMADE architecture and uses mobile technology to inform users about alerts, warnings and specific information about the daily routine of the application environment. It improves quality of the communication and control services and helps to tune up the performance of services that are implemented in architecture. The alert system provides easy-to-use configuration facilities that allow the users to select the type of information they are interested in, and to receive it immediately on their mobile phone or PDA. The information is categorized and each of the users determines the categories he is interested in. The system automatically sends updated information to the users as soon as it is available. In Figure 3, it can be seen a scheme of the system functioning.

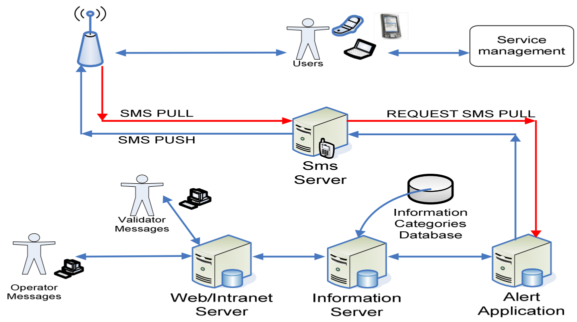

Fig. 3. Alerts Systems framework.

The alert system integrates with Java Card [16] and RFID [9] technologies, since the primary purpose of the system is to detect and prevent location and identification alarms. The system identifies objects and people, characterized with a unique serial number, using radio waves. Java Card is a technology allowing small Java applications (applets) to be run safely in microchip smart cards and similar embedded devices. Java Card gives the user the ability to program applications that can be run off a card so that it has a practical function in a specific application domain.

4.1 Alerts system features.

Users wishing to make use of this service must register, indicating his phone number and email address and give permission to receive messages. This is done through an application that implements the service. At this moment, the user can select the categories of information they wish to receive messages, in order to only receive useful and personalized information.

The system is proactive, that is, it automatically monitors the environment in which the service is implemented, trying to find information suitable to the user interest. This is very appreciated by the users, especially in home care environments, where the care personnel can receive alerts on their mobile devices, and avoid unnecessary journeys. The application automatically generates alert messages and warnings and is able to send information on time or scheduled. The application defined a number of operators and reviewers of information that are responsible for inserting and sending messages to the users. The service alerts and warnings are defined as a sub-application of the AMADE architecture, but can operate independently. The application is implemented by means of Struts. As shown in Figure 3 there are three profiles clearly differentiated, but not exclusive: users or subscribers, issuers and reviewers messages. Subscribers have to accept the conditions of the subscription service to introduce or modifying their personal data and subscribe to categories. Issuers are responsible for writing and confirming messages. Each issuer sends information only to categories assigned to it. The reviewers are responsible for validating, if necessary, the messages issued by issuers. The reviewers only review messages of assignment categories. On the other hand, as shown in Figure 3, the system handles requests for information via mobile devices, so-called SMS PULL services. These messages are sent to a special issue with a specific text, which depends on the information asked, and the system is responsible for managing and providing the information requested. The SMS PUSH service is generated automatically by the system without any user explicit request. Figure 4 shows the options available for a reviewer user.

Fig. 4. Alerts system implemented in AMADE.

4.2 Identification system, features and operation.

Java Virtual Machine (JVM) is seen as part of the operating system, called memory ROM in a smart card Java. The JVM is divided into two parts: the converter and runtime (JCRE). The converter, located in the external reader that connects to the card, makes the verification and translates byte code (compiled code) to a code inserted into the card. The JCRE, called on the card, manages the installation process, selection, deselect, execution and removal of an applet.

The main features of Java Card are portability and security; it is described in ISO 7816. Portability because the definition of standard Java Card applet allows the same function in different microchip, much like a Java applet running on different computers. The security is determined by various aspects such as an applet that is a state machine that only processes commands received via the device reader sending and responding with status codes and data. The different applications (applets) are also separated from each other by a firewall which limits access and control of data elements of a subprogram to another. Each instance of AID or applet has its unique identifier through which are selectable. The applet is responsible for their data processing commands, elements which publishes and security in the data sharing. The data are stored in the application and the Java Card applications are executed in an isolated environment, separate from the operating system and from computer that reads the card. The most commonly used algorithms, such as DES, 3DES, AES, and RSA, are cryptographically implemented in Java Card. Other services such as electronic signature or key generation are also supported.

RFID technology is grouped into the so-called automatic identification technologies. The radio frequency identification (RFID) [9] is a wireless communication technology used to identify and receive information on humans, animals or objects in motion. An RFID system consists principally of four elements: tags, readers, antennas and radios and Processing Hardware. The labels or RFID chips are passive (without batteries) and are called transponders. The transponder is located in an object (such as a bracelet), and when the transponder enters the range reading of the reader, it is activated and begins to send electromagnetic signals, transmitting its identification number (ID) the reader. The reader relays information to a central processor in which information is processed. The information is handled, it is not only restricted to data relating to the location, but it is possible to work with information on the same subject. The main applications of RFID technology have occurred in industrial environment, transport, etc. Its application in other sectors, including medicine, is becoming increasingly important. RFID provides more information than other auto-identification technologies, speeds up processes without losing reliability, and requires no human intervention. The combination of these two technologies allows us to both identify the user or identifiable element, and to locate it, by means of sensors and actuators, within the environment, at which time we can act on it and provide services. The microchip, which contains the identification data of the object to which it is adhered, generates a radio frequency signal with this data. The signal can be picked up by an RFID reader, which is responsible for reading the information and sending it, in digital format, to the specific application.

5 Case Study: Home Care Dependent Environments.

The growing elderly population imposes an urgent need to develop new approaches to care provision. Recent developments in a number of technologies, such as Muti-Agent systems, federated information management, safe communications, hypermedia interfaces, rich sensorial environments and increased intelligence of home appliances represent important enabling factors for the design and development of virtual elderly support community environments. This segment of elderly population is one of the most benefited with the development of these systems. In particular, the AMADE platform has been employed to develop a multi-agent system aimed to enhance assistance and care for low dependence patients at their homes.

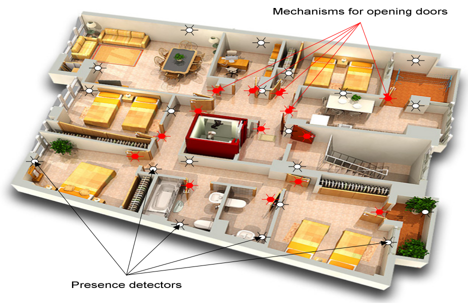

Fig. 5. Home plane.

As shown in Figure 5 are installed a series of detectors presence and mechanisms for opening doors that interact with the microchip Java Card users to offer services in real time. This sensors network through a system of alerts is responsible for generating alarms comparing the user current state with the parameters of the user daily routine who has stored the system. The system can generate alarms if it is determined the parameters for example if the user in a non-working day stands before a certain hour, or if the user spends more time than specified on the door of your home without entering, or the user is a long time motionless in the hallway, etc.

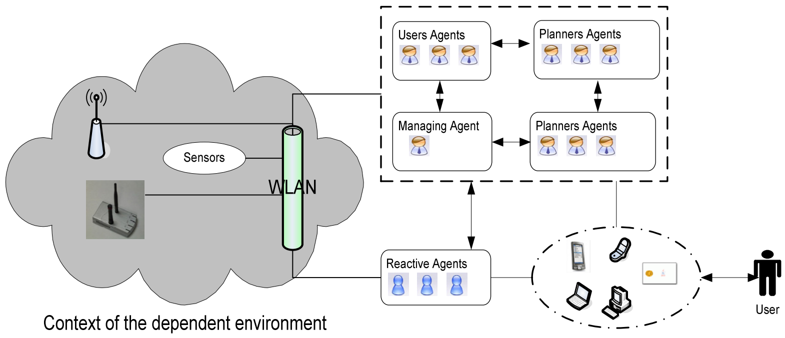

Fig. 6. AMADE structure in a dependent environment.

Main functionalities in the system include reasoning, planning mechanisms, management alerts and responses in execution time offered to certain stimuli, as shown in Figure 6. These functionalities allow the system the use of several context-aware technologies to acquire information from users and their environment. Among the technologies used are mobile systems for alerts service managing across PDA and mobile phones, Java Card elements for identification and presence detectors and access control. Each agent in the system has its own functionalities. If an agent needs to develop a task in collaboration with other agent a request form is sent. There are priority tasks that a set of agents can perform. This ensures that the priority tasks are always available.

There are four specific types of deliberative BDI [15] agents included in the case study which collaborate with the predefined agents in the AMADE platform:

- User Agent manage user personal data and their behavior. They are responsible through the system, identify and locate implemented by the architecture. They determine the status of the user and offering services in the environment as a correct temperature, automatic lighting, access blocking or opening, etc.
- SuperUser Agent runs on mobile devices and inserts new tasks into the Manager Agent to be processed by a reasoning mechanism. It also needs to interact with the User Agents to impose new tasks and receive periodic reports, and with the ScheduleUser Agents to ascertain plans' evolution.
- SheduleUser Agent schedules the users' daily activities obtaining dynamic plans depending on the tasks needed for each user. It manages scheduled-users profiles, tasks, available time and resources. Every agent generates personalized plans depending on the scheduled-user profile.
- Manager Agent runs on a Workstation and plays two roles: the physical security role that monitors the users and the manager role that handle the data and the tasks assignment for the medical staff. It must provide security for the users and ensure the tasks assignments are efficient.

On the other hand there are a number of reactive agents that work in collaboration with the deliberative agents. These agents are in change of control devices interacting with sensors (access points, lights, temperature, alarms detection, etc.). They receive information, monitor environment services and also check the devices status connected to the system. All information is treated by the reactive agent and it is sent to the manager agent to be processed.

6 Results and Conclusions.

AMADE has been used to develop a system for monitoring dependent patients at home. The testing environment has evolved from the one used for the previous ALZ-MAS architecture [7]. The ALZ-MAS architecture allows the monitoring of patients in geriatric residences, but home care is carried out through traditional methods. As mentioned in [7], ALZ-MAS present certain advantages with respect to previous architectures. However, one of the problems of ALZ-MAS was the number of agents crashed during tests. AMADE solves this problem using replicated services and reducing the crashes an average of 9%. Moreover, AMADE incorporates mobile SMS technology for managing service alerts through PDAs and mobile phones, which provides a remote alert system, and Java Card technology for identification and access control, which improves the previous RFID location technology. The environment includes reasoning and planning mechanisms, and alert and response management. Most of these responses are reactions in real time to certain stimuli, and represent the abilities that the reactive agents have in the AMADE architecture based platform. Real-time systems require an infrastructure characterized by their response time for computing and communication processes [6]. The time is considered as a critical parameter [11] to react to sensor values, i.e. when a door has to be automatically open or closed after detecting a patient identification. In this kind of situations the reactive agents receive behaviors from the deliberative agents and take control of the actions carried out.

One of the main contributions of the AMADE architecture is the remote alert system. We implemented several test cases to evaluate the management of alerts integrated into the system. This allowed us to determine the response time for warnings generated by the users, for which the results were very satisfactory, with response times shorter than those obtained prior to the implementation of AMADE. The system studies the information collected, and applies a reasoning process which allows alerts to be automatically generated, trying to avoid false alarms. For these alerts, the system does not only take response time into account, but also the time elapsed between alerts, and the user's profile and reliability, in order to generalize reactions to common situations. The results show that AMADE fits perfectly within complex systems by correctly exploiting services and planning mechanisms.

Table 1. Comparison between the AMADE and the ALZ-MAS architectures.

Factor	AMADE	ALZ-MAS
Average Response Time to Incidents(min.)	8 minutes	14 minutes
Assisted Incidents	12	17
Average number of daily planned tasks	12	10
Average number of services completed daily	46	32
Time employed by the medical staff to attend to an alert (min.)	75 minutes	90 minutes

Table 1 presents the results obtained after comparing the AMADE architecture to the previously developed ALZ-MAS architecture [7] in a case study on medical care for patients at home. The case study presented in this work consisted of analysing the functioning of both architectures in a test environment. The AMADE architecture was implemented in the home of 5 patients and was tested for 30 days. The results were promising. The data shown in Table 1 are the results obtained from the test cases. They show that the alert system improved the communication between the user and the dependent care services providers. The user identification and location system in conjunction with the alert system has helped to notably reduce the percentage of incidents in the environment under study. Moreover, in addition to a reduction in the number of incidents, the time elapsed between the generation of a warning and solution decreased significantly. Finally, due to the many improvements, the level of user satisfaction increased with the introduction of AMADE architecture since patients can live in their own homes with the same level of care as those offered at the residence.

References

1. Anastasopoulos, M., Niebuhr, D., Bartelt, C., Koch, J. & Rausch, A. (2005). Towards a Reference Middleware Architecture for Ambient Intelligence Systems. ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications.
2. Ardissono, L., Petrone, G. and Segnan, M. 2004. A conversational approach to the interaction with Web Services. Computational Intelligence, Blackwell Publishing. Vol. 20. pp. 693-709.
3. Bahadori, S., Cesta, A., Grisetti, G., Iocchi, L., Leone1, R., Nardi, D., Oddi, A., Pecora, F. & Rasconi, R. (2003). RoboCare: Pervasive Intelligence for the Domestic Care of the Elderly. In AI*IA Magazine Special Issue, January, 2003.
4. Bonino da Silva, L.O, Ramparany, F., Dockhorn, P., Vink, P., Etter, R. and Broens, T. 2007. A Service Architecture for Context Awareness and Reaction Provisioning. IEEE Congress on Services (Services 2007). pp. 25-32
5. L. Camarinha-Matos and H. Afsarmanesh, Design of a virtual community infrastructure for elderly care. PRO-VE'02 – 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises. Sesimbra, Portugal. (2002).
6. C. Carrascosa, J. Bajo, V. Julian, J.M. Corchado and V. Botti. Hybrid multi-agent architecture as a real-time problem-solving model. Expert Systems With Applications. Volumen 34. Numero 1. pp. 2-17. Elsevier.ISSN:0957-4174 (2008).
7. Corchado, J.M. , Bajo, J. , de Paz, Y. & Tapia, D. (2008). Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care.Decision Support Systems. ISSN 0167-9236.Vol 34 (2) pp. 382-396.
8. Corchado, J.M. , Bajo, J. & Abraham, A. (2008). GERAmI: Improving the delivery of health care. IEEE Intelligent Systems. Special Issue on Ambient Intelligence - Mar/Apr '08.
9. ITAA. (2004). Radio Frequency Identification. RFID...coming of age. Information Technology Association of America. http://www.itaa.org/rfid/docs/rfid.pdf.
10. Kleindienst, J., Macek, T., Seredi, L., & Sedivy, J. (2004). Vision-enhanced multi-modal interactions in domotic environments. IBM Tecnologías de Voz y Sistemas. República Checa.
11. Jacobsen V., Fredrick R., Casner S., and Schulzrinne H., "RTP: A transport Protocol for Real-Time Applications". RFC 1889. Lawrence Berkeley National Laboratory, Xerox PARC, Precept Software Inc., GMD Fokus, January 1996. Online. Internet. http://www.connect.org.uk./teckwatch/cgi-bin/rfcshow?1889 Accessed 16 Oct., 1996
12. Julian, V., Botti, V.: Developing real-time multi-agent systems. In: Proceedings of the Fourth Iberoamerican Workshop on Multi-Agent Systems (Iberagents'02), Málaga (2004).
13. Ricci, A., Buda, C. and Zaghini, N. 2007. An agent-oriented programming model for SOA & web services. In 5th IEEE International Conference on Industrial Informatics (INDIN'07), Vienna, Austria. pp. 1059-1064.
14. Richter, K. & Hellenschmidt, M. (2004). Interacting with the Ambience: Multimodal Interaction and Ambient Intelligence. Position Paper to the W3C Workshop on Multimodal Interaction, 19-20 July 2004.
15. Tapia, D. I., Bajo, J., De Paz, F. & Corchado, J. M. (2006). Hybrid Multiagent System for Alzheimer Health Care. Proceedings of HAIS 2006. Solange Oliveira Rezende, Antonio Carlos Roque da Silva Filho Eds. Ribeirao Preto, Brasil.
16. ZhiqunChen (Sun Microsystems). Java Card Technology for Smart Cards. Addison Wesley Longman. ISBN 0201703297.