Inteligencia Artificial 42 (2009) 36-45
doi: 10.4114/ia.v13i42.1044

INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

# Multi-Agent Architecture for Dependent Environments. Providing Solutions for Home Care.

[1]Juan A. Fraile, [1]Javier Bajo and [2]Juan M. Corchado
[1]Universidad Pontificia de Salamanca. Compañía 5, 37002, Salamanca, Spain
{jafraileni;jbajope}@upsa.es
[2]Departamento de Informática y Automática. Universidad de Salamanca. Plaza de la Merced s/n, 37008, Salamanca, Spain
corchado@usal.es

**Abstract** Obtaining novel and effective solutions for home care is one of the aims of Ambient Intelligence. Elderly and disabled people require special care services, and emerging technologies facilitate the development of intelligent environments in 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. This study proposes a new perspective in tackling the problem of home care. An initial prototype based on the AMADE architecture was 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 [7]. In recent years, there has been an important growth in the field of Ambient Intelligence (AmI) [1] [18], involving major changes in the daily lives of people. The vision of Ambient Intelligence involves the creation of intelligent spaces where users interact in a natural way with computational systems and communication technologies, which are invisible and ubiquitous. The technology is adapted to individuals and their context, acting autonomously, and facilitating their daily tasks. One of the main goals of Ambient Intelligence focuses on building systems that support daily life activities, including those related to home automation [12], in an efficient way.

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 resource management. Additionally, most of the current systems require concurrent work to provide real time solutions. This paper presents Architecture for 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, such as patient monitoring and tracking, and provide a quick response to problematic situations. Because of their particular characteristics [19], multi-agent systems are very appropriate for satisfying these needs. One of the main contributions of AMADE is the use of both reactive and deliberative agents, which facilitates advanced reasoning capabilities and real-time reactive behaviors. These are two important characteristics that must be taken into account in the development of intelligent environments.

**ISSN: 1137-3601(print); 1988-3064(on-line)**

This paper is organized as follows: the section two 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 the architecture applied to a real case and the last section presents the results and conclusions obtained.

## 2 General description of the problem

Home Care is one of the priorities of Ambient Intelligence, since dependent people require new solutions that can make use of technological advances to provide novel and fundamental care services [2]. The vision of Ambient Intelligence is to improve the quality, equity and continuity of health care [2]. In this sense, an intelligent environment can improve health care services and can have a high social impact, especially in the home care services for dependent chronic patients [3]. Home Care requires effective communication and remote monitoring [3]. Multi-agent systems facilitate the development of home care environments. Agents are autonomous software entities [7] able to interact with their surroundings, and highly capable of adapting to changes. Agents can communicate with other agents and work in a coordinated manner. Moreover, the agent-oriented methodologies provide a mechanism for modelling distributed, inter-operable and secure systems by taking social and organizational considerations into account, and integrating multiple devices, sensors and humans.

The use of intelligent agents is an essential component for analyzing information from distributed sensors [19] [15] [16]. 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 [5]. Although multi-agent systems already exist and are capable of gathering information within a given environment in order to provide medical care [9], 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. With 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 [10].

Hybrid architectures try to combine deliberative and reactive aspects by combining deliberative and reactive modules [8]. The reactive modules are in charge of processing stimuli that do not 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 multi-agent systems, which can help distribute resources and 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 [4]. 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 [6] [17]. 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 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 users subscribed in the system that have requested such alerts. The moment at which the alert is sent depends on the system parameters and the urgency of the warning. In order to identify each user, AMADE implements a system based on Java Card [20] and RFID [11] (Radio Frequency IDentification) microchip technology with a series of distributed sensors that provide the user with the necessary services.

## 3 AMADE Multi-Agent Architecture.

AMADE is multiagent architecture specifically designed to facilitate the development of intelligent environments for home care. The AMADE architecture incorporates intelligent agents, identification and localization technologies, wireless networks and mobile devices. Additionally, it provides access mechanisms 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. User identification and location facilities in AMADE are based on Java Card and RFID technologies. AMADE takes flexibility and extensibility requirements into account. These requirements are common to all kinds of distributed systems, but especially apply to this application domain because of its intrinsic characteristics, such as 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.

The AMADE architecture is composed of four basic blocks that can be seen in Figure 1: Applications, Services, Agent Platform and Communication Protocol. These blocks constitute the complete functionality of the architecture. AMADE allows the use of devices such as PDAs, mobile phones, laptops, sensors or Java Card chip with RFID technology. The devices used in the AMADE architecture execute applications or services and provide users with different functionalities. The applications often require the execution of services, which they request through the agent platform. The agent platform consists of deliberative agents and reactive agents. Reactive agents are responsible for monitoring sensors and actuators, while deliberative agents are responsible for monitoring services and applications in the architecture. The reactive agents and the deliberative agents function interactively, with the former providing context information and the latter receiving tasks to execute. The agent platform controls the security, communications and the workflow in AMADE. A multiagent-based architecture design facilitates the flexibility of adding new agents at any time with new features to the system.

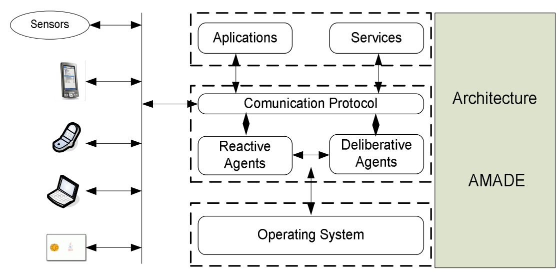


Figure 1. AMADE Framework.

The agent 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. The first group of agents is made up of deliberative BDI agents, which are in charge of the management and coordination of all the 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, and they can be executed on mobile devices, so they are always available and they provide ubiquitous access for the users. There are different kinds of deliberative agents in the architecture, each one with specific roles, capabilities and characteristics. This provides the architecture with the flexibility to incorporate new agents.

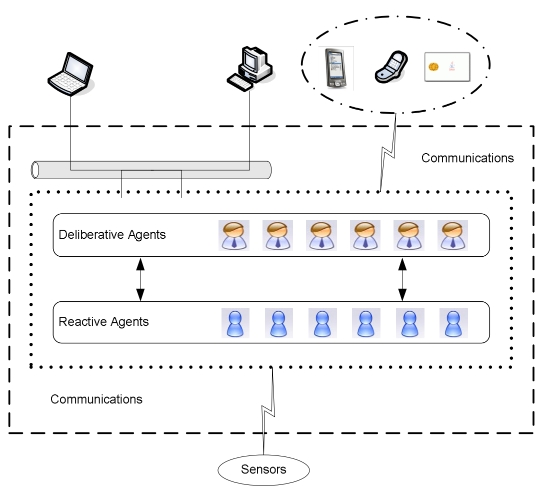


Figure 2. Agent platform structure in the AMADE architecture.

Specific pre-defined agents provide the basic functionalities of the architecture:

- CoAp Agent: This agent is responsible for all communication between the applications and the platform. It 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 a service.
- CoSe Agent: It is responsible for all communication between services and the platform. The functionalities are similar to CoAp Agent but in reverse. This agent is always on listening mode waiting for responses of services. Manager Agent lets CoSe Agent know which service 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 periodically verifies the status of all agents registered in the architecture by sending ping messages. If there is no response, the supervisor agent kills the system 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 by taking user preferences into account. Users can explicitly invoke a service, or let the Manager Agent decide which service is best 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 a higher and better performance than the others. Manager Agent has a routing list to manage messages from all applications and services.
- Interface Agent: This agent is designed to be embedded in user applications. Interface agents communicate directly with the agents in AMADE so there is no need to employ any communication protocol other than FIPA ACL. The requests are sent directly to the Security Agent, which analyzes the requests and sends them to the Manager Agent.

The second group of agents on the platform 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 do not 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 [14]. 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 a 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 agent platform. The protocol obtained 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) [13] [8] 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 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 Location, Identification and Alert Systems in AMADE

The alert system is integrated into the AMADE architecture and uses mobile technology to inform users about alerts by sms, warnings and information specific to the daily routine of the application environment. It improves the service quality of communication, and constantly controls the performance of applications that are implemented in AMADE architecture.

This is a very configurable system that allows 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 the users determine the categories they want to subscribe to. The system automatically sends the information to each of the users as soon as it is available. To use a service, interested users must register, indicating their phone number and email address, and permit messages to be received. Figure 3 illustrates how the system functions.

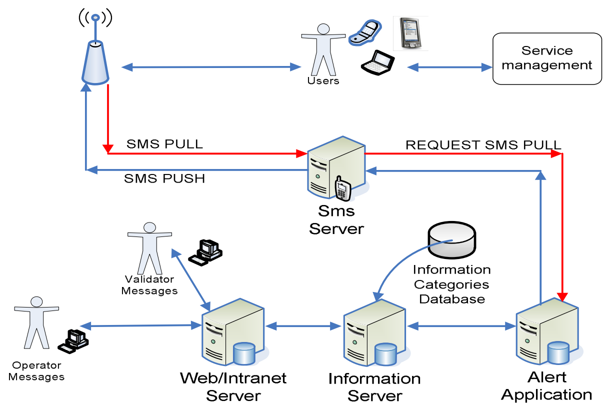


Figure 3. Alerts Systems framework.

The alert system integrates Java Card [20] and RFID [11] technologies, since the primary purpose of the system is to detect and prevent location and identification alarms. The system uses radio waves to identify objects and people, which are characterized with a unique serial number.

### 4.1 Alerts system features.

The alerts system is proactive in that it automatically contacts a user when new information is available. As such, users can have updated information on their personal devices immediately and ubiquitously, avoiding temporal and spatial restrictions. Moreover, the alerts system facilitates the scheduling of pre-programmed alerts and warnings, which is very useful, for example, for remembering to take medications. Several reviewers are defined to revise the contents of the messages and to authorize the information, specifically when working with critical data or advice.

As shown in Figure 3, there are three profiles that are clearly distinct, but not exclusive of each other: users or subscribers; issuers; and message reviewers. Subscribers need to accept the conditions of the subscription service to introduce or modify their personal data and to subscribe to desired categories. Issuers are responsible for writing messages and confirming subsequent validations. Each issuer can only request or send information to the categories that it has previously subscribed to. The reviewers are responsible for validating, if necessary, the messages written by the issuers. Each of the reviewers is assigned to one or more categories and is denied access to the rest of categories. The web/intranet server is the application responsible for managing the alerts information. The information server defines information categories, user profiles and information to be sent. The alerts application communicates with the information server for pending shipments and to consult information requested by the users. Finally, the SMS server is responsible for distributing information through mobile operators.

As shown in Figure 3, users can request information to the alerts system via mobile devices, using the SMS PULL service. These messages are sent to a special issuer and contain a specific text, depending on the information demanded. The system filters the key text in the SMS message and provides the information requested. The SMS PUSH is automatically generated by the system without the user's explicit request.

## 4.2 Location and Identification System.

The location and identification system in AMADE incorporates Java Card and RFID technologies. The primary purpose of this system is to convey the identity of an object or person, associated to a unique serial number, using radio waves. Java Card is a technology that permits 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.

Java Virtual Machine (JVM) is 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, verifies and translates byte code (compiled code) to a code inserted into the card. The JCRE manages the installation process, selection, deselection, execution and removal of an applet.

The main features of Java Card are portability and security, as described in ISO 7816. Portability is essential because the definition of a standard Java Card applet allows the same function in different microchips, much like a Java applet running on different computers. The security is determined by various aspects such as an applet, which is a state machine that only processes commands received via the device reader by 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 the data elements of one subprogram to another. Each instance of AID or applet has its unique identifier. The applet is responsible for data processing commands, elements which it publishes, and data sharing security. 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 the 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 forms part of the so-called automatic identification technologies. The radio frequency identification (RFID) 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 reading range of the reader, it is activated and begins to send electromagnetic signals, transmitting its identification number (ID) to the reader. The reader relays information to a central processor in which information is processed. The main applications of RFID technology have occurred in the industry and transportation sectors. 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.

The location and identification system is coordinated by the agent platform. The reactive agents work in collaboration with the deliberative agents. These agents are in charge of control devices that interact with sensors (access points, lights, temperature, location, identification, etc.). They receive information, monitor environment services and also check the status of devices connected to the system. All information is handled by the reactive agent and is sent to the manager agent to be processed.

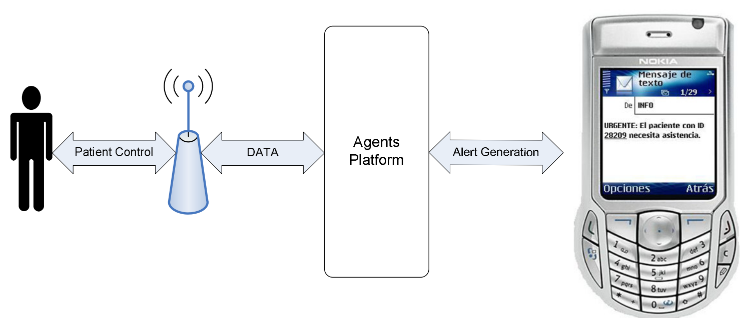


Figure 4. Alert SMS example sent by the system.

Figure 4 illustrates the coordination between the alerts system and the location and identification system. In this example, a warning is generated in the system when a patient has remained motionless in the hallway at home longer than usual. The RFID sensors determine the patient’s location, and a reactive agent detects the anomalous situation. This agent informs the patient’s personal agent about the possible risk situation. The personal agent detects the warning situation and sends a special message to the Manager agent. The Manager agent is the deliberative agent in charge of managing risky situations, and makes the decision to initiate the alarm procedure.

The Manager agent takes past similar experiences into account, evaluates the current situation and detects a potential risk. Then it generates an alarm message that it encapsulates in a SMS and delivers to the corresponding care givers and the patient's family. The warning and the real diagnosis of the situation are stored in the Manager agent's cases memory for future similar situations.

# 5 Case Study: Home Care Dependent Environments.

The AMADE architecture was specifically employed to develop a multi-agent system aimed at enhancing assistance and care for low dependence patients at their homes. A case study was developed at a house measuring 89 m² and specifically constructed for two dependent people. As shown in Figure 5, 20 passive SX-360 infrared motion detectors were installed in the house, as well as 11 automatic door opening mechanisms. These mechanisms interact with the Java Card & RFID microchip carried by the users. Each disabled user is identified by a Sokymat ID Band Unique Q5 bracelet, which integrates a RFID-Java-Crypto-Card chip with a 32K Module and Crypto-CoProzessor (1024 bit RSA) compatible with the SUN JavaCard 2.1.1. The sensors and the actuators are placed in strategic positions within the home as shown in Figure 5. As explained in Section 4, each of these physical devices is controlled by reactive agents. The reactive agents connect the sensor network to the alert system and provide a framework for monitoring the two disabled persons and for generating alarms when detecting risky situations. The system compares the user's current state with the parameters of the user's daily routine and generates an alarm when the parameters observed do not match the normal parameters. Some examples are if the user gets up prior to a specific hour on a non-work day, if the user spends more time than normal standing at a door without entering, or if the user remains motionless in the hallway for an extended period of time.

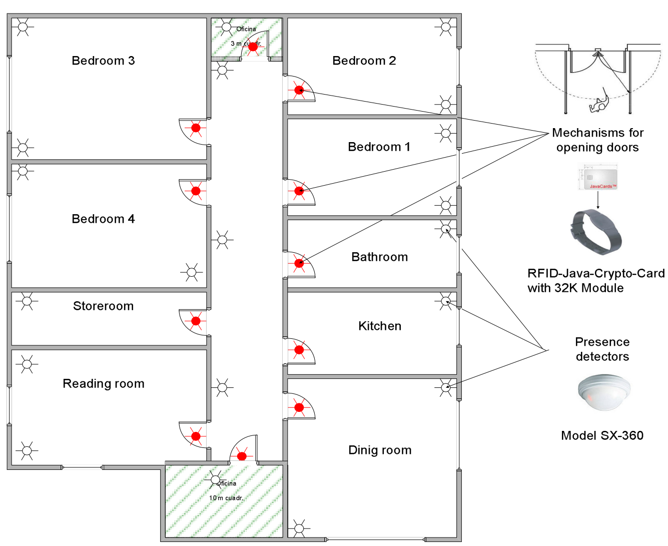


Figure 5. Home plan.

The main functionalities in the system include reasoning, planning mechanisms, alerts management and responses in execution time offered to certain stimuli, as shown in Figure 6. These functionalities require the use of several context-aware technologies to acquire information from users and their environment. Among the technologies used in the case study are mobile devices, such as PDAs and mobile phones, Java Card elements for identification, presence detectors, and access control. Each agent in the system has its own functionalities. If an agent needs to develop a task in conjunction with another agent, a request form is sent. There are priority tasks that a set of agents can perform, which ensures that priority tasks are always available.

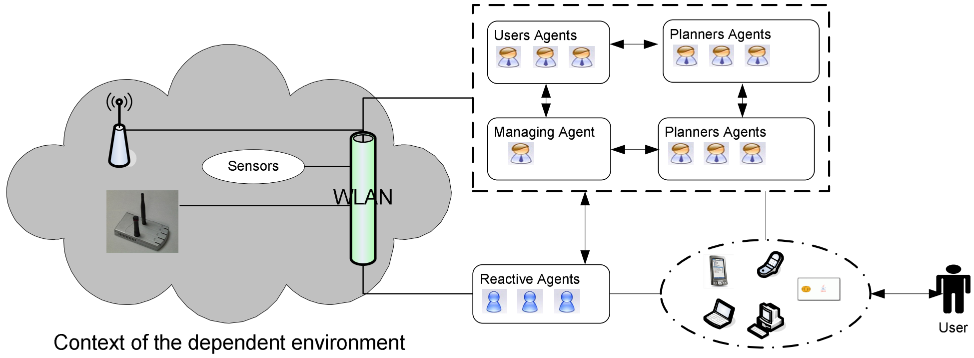


Figure 6. AMADE structure in a dependent environment.

There are four specific types of deliberative BDI [19] agents included in the case study which collaborate with the predefined agents in the AMADE platform:

- User Agent manages the user's personal data and behavior. They determine the status of the user and offer services related to the environment, such as a maintaining 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 the evolution of the intended plans.
- SheduleUser Agent schedules the users' daily activities and obtains dynamic plans depending on the tasks needed for each user. It manages scheduled-user 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 user, and the manager role that handles the data and the task assignments for the medical staff. It must provide security for the users and ensure the task assignments are efficient.

There are additionally a number of reactive agents that work in conjunction with the deliberative agents. These agents are in charge of control devices that interact with sensors (access points, lights, temperature, alarms detection, etc.). They receive information, monitor environment services and also check the status of devices connected to the system. All information is handled by the reactive agent and is sent to the manager agent for processing.

# 6 Results and Conclusions.

AMADE was used to develop a system for monitoring disabled patients at home. The testing environment has evolved since the previous study which used ALZ-MAS architecture [9]. The ALZ-MAS architecture can monitor patients in geriatric residences, but home care is provided through traditional methods. As mentioned in [9], ALZ-MAS present certain advantages with respect to previous architectures. However, one of the problems of ALZ-MAS was the number of agents that crashed during tests. AMADE solves this problem by using replicated services, which reduced the number of crashes by 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. It also incorporates 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 their communication processes [8]. Time is considered a critical parameter [13] to react to sensor values, e.g., when a door has to be automatically opened or closed after a patient's identification has been detected. In this kind of situation the reactive agents receive behaviors from the deliberative agents and take control of the actions to be 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, and tries to avoid false alarms. For these alerts, the system does not only take response time into account, but also the time elapsed between alerts, as well as 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.) | 3 minutes | 6 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 [9] in a case study on medical care for patients at home. The case study presented in this paper 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 the eventual 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 that offered at a care home.

The use of multi agent technique for implementing the system can be enhanced in the future in such a way that, the agents can be enhanced by adding more intelligent tasks to its jobs to enhance the system's performance. We can also increase the intelligence of the implemented agents such as the select agent and the help agent to give more supporting to the patients.

## Acknowledgements

This development has been supported by the projects JCYL - SA071A08, MITYC - ABANSA-TSI-020100-2008-276 and MITYC - SIAAD-TSI-020100-2008-307.

## References

[1] M. Anastasopoulos, D. Niebuhr, C. Bartelt, J. Koch and A. Rausch. Towards a Reference Middleware Architecture for Ambient Intelligence Systems. In *Proc. of the Workshop for Building Software for Pervasive Computing, 20th Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA)*, pages 145-157, 2005.

[2] C. Angulo and R. Tellez. Distributed Intelligence for smart home appliances. *Tendencias de la minería de datos en España.* Red Española de Minería de Datos*,* 2004.

[3] J. C. Augusto and P. McCullagh. Ambient Intelligence: Concepts and Applications. *International Journal on Computer Science and Information Systems,* 4(1):1-28, 2007.

[4] L. Ardissono, G. Petrone and M. Segnan. A conversational approach to the interaction with Web Services. *Computational Intelligence,* 20(4):693-709, 2004. DOI: http://dx.doi.org/10.1111/j.0824-7935.2004.00261.x

[5] S. Bahadori, A. Cesta, G. Grisetti, L. Iocchi, R. Leone1, D. Nardi, A. Oddi, F. Pecora and R. Rasconi. RoboCare: Pervasive Intelligence for the Domestic Care of the Elderly. *Intelligenza Artificiale*, 1(1):16-21, 2004.

[6] L. O. Bonino da Silva, F. Ramparany, P. Dockhorn, P. Vink, R. Etter, and T. Broens. A Service Architecture for Context Awareness and Reaction Provisioning. *IEEE Congress on Services,* pages 25-32, 2007. DOI: http://dx.doi.org/10.1109/SERVICES.2007.10

[7] L. Camarinha-Matos and H. Afsarmanesh. Design of a virtual community infrastructure for elderly care. In proc. of *3rd IFIP Working Conference on Infrastructures for Virtual Enterprises: Collaborative Business Ecosystems and Virtual Enterprises,* pages *635,* 2002.

[8] 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,* 34(1):2-17, 2008.

[9] J.M. Corchado, J. Bajo, Y. de Paz, and D. Tapia. Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care. *Decision Support Systems,* 34(2):382-396, 2008. DOI: http://dx.doi.org/10.1016/j.dss.2007.04.008

[10] J.M. Corchado, J. Bajo and A. Abraham. GERAmI: Improving the delivery of health care. *IEEE Intelligent Systems,* 3(2):19-25, 2008. DOI: http://dx.doi.org/10.1109/MIS.2008.27

[11] ITAA. Radio Frequency Identification. RFID... coming of age. *Information Technology Association of America,* 2004. http://www.itaa.org/rfid/docs/rfid.pdf.

[12] J. Kleindienst, T. Macek, L. Seredi and J. Sedivy. Vision-enhanced multi-modal interactions in domotic environments. In *Proc. of the 8th ERCIM Workshop "User Interfaces For All",* 2004.

[13] V. Jacobsen, R. Fredrick, S. Casner, and H. Schulzrinne. RTP: A transport Protocol for Real-Time Applications. RFC 1889, 1996. Online: http://www.connect.org.uk./teckwatch/cgi-bin/rfcshow?1889

[14] V. Julian and V. Botti. Developing real-time multi-agent systems. *Integrated Computer-Aided Engineering*. 11(2):135-149. 2004.

[15] J.M. Molina, J. García, F.J. Jiménez and J.R. Casar. Cooperative Management of a Net of Intelligent Surveillance Agent Sensors, *International Journal of Intelligent Systems,* 18(3):279-307, 2003. DOI: http://dx.doi.org/10.1002/int.10089

[16] J. Pavón, J. Gómez Sanz, A. Fernández Caballero and J. J. Valencia Jiménez. Development of intelligent multi-sensor surveillance systems with agents. *Robotics and Autonomous Systems,* 55(12):892-903, 2007. DOI: http://dx.doi.org/10.1016/j.robot.2007.07.009

[17] A. Ricci, C. Buda, and N. Zaghini. An agent-oriented programming model for SOA & web services. In *Proc. of 5th IEEE International Conference on Industrial Informatics (INDIN'07)*, pages 1059-1064, 2007.

[18] K. Richter and M. Hellenschmidt. Interacting with the Ambience: Multimodal Interaction and Ambient Intelligence. In *Proceedings of the W3C Workshop on Multimodal Interaction,* 19:20, 2004.

[19] D.I. Tapia, J. Bajo, F. De Paz and J.M. Corchado. Hybrid Multiagent System for Alzheimer Health Care. In *Proc of Hybrid Artificial Intelligence Systems Workshop HAIS'06*, pages 121-130, 2006.

[20] Z. Chen. *Java Card(TM) Technology for Smart Cards: Architecture and Programmer*, Prentice Hall, Boston, USA, 2000.