HoCaMA: Home Care Hybrid Multiagent Architecture

Juan A. Fraile and Javier Bajo and Ajith Abraham and Juan M. Corchado

Abstract Home Care is one of the main objectives of Ambient Intelligence. Nowadays, the disabled and elderly population, which represents a significant part of our society, requires novel solutions for providing home care in an effective way. In this chapter we present HoCaMA, a hybrid multiagent architecture that facilitates remote monitoring and care services for disabled patients at their homes. HoCaMA combines multiagent systems and Web services to facilitate the communication and integration with multiple health care systems. In addition, HoCaMA focuses on the design of reactive agents capable of interact with different sensors present in the environment and incorporates a system of alerts through SMS and MMS mobile technologies. Finally, it uses RFID and JavaCard technologies to provide advanced location and identification systems as well as automatic access control facilities. HoCaMA has been implemented in a real environment and the results obtained are presented within this chapter.

Key words: Ambient Intelligence, Home Care, Multiagent System, Ubiquitous Computing

Juan A. Fraile
Pontifical University of Salamanca, Compañía 5, 37002, Salamanca, Spain e-mail: jafraileni@upsa.es

Javier Bajo
Pontifical University of Salamanca, Compañía 5, 37002, Salamanca, Spain, e-mail: jbajope@upsa.es

Ajith Abraham
Norwegian University of Science and Technology, e-mail: ajito.abraham@ieee.org

Juan M. Corchado
University of Salamanca, Plaza de la Merced s/n, 37008, Salamanca, Spain, e-mail: corchado@usal.es

1 Introduction

Nowadays, there is a considerable growth in the development of automation technologies as well as intelligent environments as demonstrates the relevance acquired by the Ambient Intelligence (AmI) [1], [9]. One of the main objectives of AmI is to look after the user's well-being and obtain friendly, rational, productive, sustainable and secure relationship for users within their environments. Several architectures based on multiagent systems have emerged thanks to their appropriateness to be applied in the development of intelligent spaces and the integration of devices that are programmable via computer networks [23]. Moreover, they contribute to the development of ubiquitous computation [15], which notably helps to resolve the challenge of defining strategies that can allow the early detection and prevention of problems in an automated environment.

The ubiquitous environments require the improvement of the services offered to the users as well as the way they can be accessed. Moreover, it is necessary the adoption of trends already tested and proven in technological environments. Ubiquitous environments are focused on the user, since he is the center of the new technological facilities and demands access to unified services. In this sense, every user interacts in a personal way for the enjoyment of services. An ubiquitous environment is constituted by devices sensitive to the presence of the user and is able to automatically identify and recognize the user preferences. There are several benefits provided by Ambient Intelligence services [22]: users get easier access to services that are contracted, access to services is independent of the terminal which is used and use of services is simpler, allowing a rapid assimilation by the user. In addition, users can receive personalized services entirely, so they have quick access to what they call their personal needs.

The high growth of the sector of the population requiring Ambient Intelligence solutions in home care environments, as well as commitments that are being acquired to meet the needs of this sector, suggest that it is necessary to modernize the current solutions. Multiagent systems [34] and architectures based on the use of intelligent devices, have been recently explored as monitoring systems for medical care services [2] for the elderly and people with Alzheimer's disease [9]. Multiagent systems can provide continued support in the daily life of disabled people [7], predicting potentially dangerous situations and hanging over a cognitive and physical support to the medical staff [4].

This work uses the possibilities provided by the new technologies to develop an open multiagent architecture, which can integrate video surveillance techniques, artificial intelligence, intelligent agents, wireless and mobile technology. The use of these technologies facilitates new functionalities that allow optimizing the management and efficiency of services and applications, improving the quality of life of users and facilitating the work of the care givers.

The main objective of this paper is to define a hybrid MultiAgent architecture, HoCaMA, for the control and the supervision of open environments. HoCaMA involves functionalities that allow automated identification, localization, alarms management and control of movement. The users of solutions based on HoCaMA are be able to gain wireless access to all the information that they need to perform their work. A novel real time communication protocol at the core of the architecture allows secure and rapid communication between the reactive agents and the system sensors. The reactive agents, whose response time is critical, are influenced by deliberative BDI agents, which are located inside the platform. Additionally, the HoCaMA architecture integrates an alert system across the agents' platform specifically designed to work with mobile devices. The alert system contains different levels of emergency, and the alert level is determined by a deliberative agent who, depending on the alert level, emits the alarm to the appropriate agent, either a reactive agent or a deliberative agent.

The paper is organized as follows: The second section presents the problem that prompted this work. The third section presents the proposed architecture, and the fourth section presents as the architecture applies to real case. The fifth section gives the results and conclusions obtained after applying the proposed architecture to a real case in an environment of dependence.

2 General Description of the Problem

The use of intelligent agents is an essential component for analyzing information on distributed sensors [27], [33]. 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 [4]. Although multiagent systems already exist and are capable of gathering information within a given environment in order to provide medical care [18], 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 [10]. The PROSAFE project [8] attempts to automatically identify the daily activities of the monitored person. The processing of collected data is carried out on doctor's request with an adapted interface. The final operational objective is to detect any abnormal behaviour such as a fall, a runaway, or an accident.

The research objective is to gather characteristic data about the nightly or daily activities of the patient. The AILISA project [20] (Intelligent Apartments for effective longevity) is an experimental platform to evaluate remote care and assistive technologies in gerontology. This ambitious project regroups specialists of smart homes, networks and computing, electronics, and signal processing. The e-Vital project [24] (cost-effective health services for interactive continuous monitoring of vital signs parameters) is a modular and ambulatory telemedicine platform. Its objective is to increase patient's feeling of safety concerning their health. Patients and caregivers feed a central database with some measuring equipment. The developed device allows staff to take measurements and data collected to be sent to the resident doctor. This doctor can remotely diagnose whether there is a problem that needs them to visit or that requires the resident to receive hospital treatment. By way of a personal digital assistant (PDA), the e-Vital server connects monitoring devices produced by several manufacturers. The server is a multiagent system where each agent focuses on a specific task related to the medical stored data. For example, an alert manager is specialized people in their own homes) is a multiagent framework in which agents use a restricted cooperation protocol to collectively perform classifications.

We chose a multiagent approach because these systems proved their adequacy in many health problems. Multiagent architecture is particularly adequate if the problem-solving implies the coordination of various specialized people (e.g., units of a hospital must collaborate to establish patient scheduling). Then, the agents have cooperative skills to communicate and to build together a solution. Moreover, many medical problems are complex and often standard solutions are not easy to find. A multiagent problem-solving is based on decomposition in sub-problems. Multiagent technology also proved its reliability in medical information retrieval.

2.1 The Pervasive Services in Home Care

According to Mark Weiser [32] more entrenched technologies are those that disappear, are intermingled in the reality of everyday life to become invisible. Weiser's motivation was to find a way to make easier the use of computers. Today, the main purpose of pervasive computing is to improve the user experience when it comes to interact with computer technology. Weiser said that pervasive systems refer to a mixture of digital and physical environments that are inhabited by computer components and communications, and integrated seamlessly with human users and their needs. Such systems makes the technology is transparent and allows users to concentrate on their tasks and experience. When designing pervasive services it is necessary to offer them so that users can operate in a manner enjoyable. Pervasive systems current can be found in environments such as automobiles, offices, public buildings and, of course, in our homes. The requirements that must be met such systems cover a multitude of services [30] in areas such as multimedia, communications or automation, so that these systems use hardware devices to provide these services. Roy Want [31] describes as mobile devices are becoming brokers of the person and their activities, improving their efficiency in the movement. A mobile phone is experiencing almost all physical parameters that may experience a person travels at the same speed, is exposed to the same temperature, the same sounds, pollution levels and is close to the same people and equipment that its owner. Therefore through a network of sensors can be registered at all the mobile owner experiences and act accordingly.

The users can interact with the pervasive environment using different devices (PDAs, laptops, PCs, mobile, etc.). Moreover, the environment can modify its behavior depending on the user context. As the computing and communication capabilities of the environment increases, newer service lines in mobility, either through wired or wireless access [26]. Looking to give users the best experience possible, all devices which are surrounded must be transformed or abstracted into a communications environment that can be accessed easily and that could be used by someone without technological experience. This requires you to hide the terminal capabilities complexity as well as the communications variety, therefore promoting the acceptance of new terminals and new services for users, without any prior training phase. Taking into account the wide variety of communication tools available today and the diversity of contexts in which they can be used there are innovative services that can respond to users' expectations [19]. These new and innovative services have to be smart enough:

- Understanding the context linked to a person or group of persons.
- Behave accordingly, both reactively (the service detects a change of context and adapts) as a proactive (service predicts something that user does not know yet and intends to adapt their behavior).
- Exploiting optimally communications capabilities available to the user.
- It has to be easy to use.

To implement all these concepts, pervasive services should behave according to the scheme described in Figure 1. Under this scheme, services are able to learn the context of the user through a series of concepts such as location, presence, mood, sensor networks, type of terminal access, etc. Moreover, users define a profile based on their preferences [19]. By combining information on the context and the user profile, obtained a series of rules that serve as input to the personalization process, where the service is provided to user so personalized and contextualized [11].

The blocks that define the pervasive services behavior are:

- The block user context. One of the main information sources to customize services to users comes from their environment or context. The term ”context awareness” refers to the ability of services to capture information from very diverse nature that is in the user environment. Some of the information sources that constitute the user context are:
  - The location. An important information data that can determine the service behavior is the place where the user at any given moment.
  - The presence / Mind state. The ability to report on our state is known as presence, even though the user himself determines which this information. This information can be enriched with details on our mood in a pervasive services context, which may come as determined by various sensors that analyze our behavior and properties of the environment in which we find ourselves.
  - The access terminal. Clearly, the terminal or terminals through which we access to services largely determine the format, quantity and quality of content they can offer.
  - The sensors networks. The characteristics of our environment can be picked up by sensors networks, which revolutionize the way to understand and manage complex contexts. The terminals interact with the environment to ob-

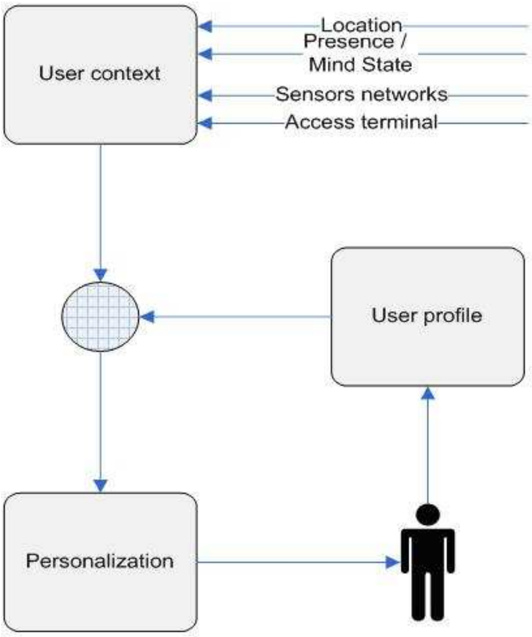


Fig. 1 Pervasive services behaviour

tain information from him. They process and store without user participation, which only delivered the data abstract, which leads to a better understanding of environment and better alignment of services to the context in which they are developing.

- The user profile block. Another information source in which a ubiquitous service must be supported to adapt to the user is the user's profile. This profile includes information that facilitates the user himself, classified into categories on their preferences and habits. Logically, the user profile information is very sensitive and must be adequately protected, so as to ensure its confidentiality. This leads us to the need to users identify, which may be done using microchip.
- The personalization block. Combining information from the proper context with the profile user, you get a set of rules that allow customized services to the environment and preferences user, or even a combination of both. Some of the actions which can be carried out during this services personalization process are adapting graphical interfaces, the content adaptation and discovery and services composition.

2.2 Multiagent Architecture Art State

At present, in terms of data transmission and security, there are many problems to solve, especially derived from the technology used in their development. However, even with these limitations, the potential of wireless systems in combination with agents and multiagent systems are unlimited. The multiagent systems can provide added value to all wireless services that will be generated with the third generation of mobile telephony, so that a user can interact with a ”personal agent” that will be available 24 hours a day. The current technology provides the necessary basis for the development of these systems.

Hybrid architectures try to combine deliberative and reactive aspects, by combining reactive and deliberative modules. 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 services centralization adversely affects the system functionality by limiting their capabilities. The classical architectures are characterized by modularity and structure that it is oriented system that applies.

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 the SOA architectures are the multiagent systems, which can help to distribute resources and to reduce the centralization of tasks. Unfortunately the complexity of designing multiagent architecture is great since there are not tools to either help programme needs or develop agents.

Multiagent systems combine aspects of both classic and modern architectures. The integration of multiagent systems with SOA and web services has been recently investigated [3]. Some investigators focus on the communication among both models, whereas others focus on the integration of distributed services, especially web services, in the agents’ structure [5], [21], [29]. Bonino da Silva et al. (2007) [5] propose to merge multiagent techniques with semantic web services to enable dynamic, context-aware service composition. They focus on SOA to design a multiagent service composition as an intelligent control layer, where agents discover services and adapt their behaviour and capabilities according semantic service descriptions. Ricci et al. (2007) [21] have developed a java-based framework to create SOA and Web Services compliant applications, which are modelled as agents. Communication between agents and services is done using what they call ”artifacts” and WSDL (Web Service Definition Language). Shafiq et al. (2006) [25] propose a gateway that allows interoperability between Web Services and multiagent systems. This gateway is an agent that integrates FIPA (Foundation for Intelligent Physical Agents) and W3C (The World Wide Web Consortium) specifications, translating ACL (Agent Communication Language), SOAP and WSDL messages, and combines both directories from agents’ platforms and web services. Li et al. (2004) [16] propose a similar approach, but focusing on the representation of services. They use SOAP and WSDL messages to interact with agents. Liu (2007) [17] proposes a multiagent architecture to develop inter-enterprise cooperation systems using SOA and Web

Services components and communication protocols. Walton (2006) [29] presents a technique to build multiagent systems using Web Services, defining a language to represent the dialogs of agents. There are also frameworks, such as Sun's Jini and IBM's WebSphere, which provide several tools to develop SOA-based systems. Jini uses Java technology to develop distributed and adaptive systems over dynamic environments. Rigole et al. (2002) [23] have used Jini to create agents on demand into a home automation system, where each agent is defined as a service in the network. Web-Sphere provide tools for several operating systems and programming languages. However, the systems developed using the mentioned frameworks are not open at all because the framework is closed and services and applications must be programmed using a specific programming language that support their proprietary API's.

The works revised in this section provide a good base for the development of multiagent systems. However, because the majority of them are in the development stage, their full potential in a real environments is unknown. HoCaMA 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 multiagent systems, thus allowing the agents and systems to function as services. Another feature is security HoCaMA that it is managed by agents. All communications are through agents, services can not share resources unless you allow the agents. The services must be defined always available and are not shared with other systems unless otherwise specified. HoCaMA 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. In order to identify each user, HoCaMA implements a system based on Java Card [34] y 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 Proposed Architecture

The HoCaMA architecture uses a model with a series of components to offer a solution that includes different levels of service for heterogeneous systems. The model incorporates intelligent agents, identification and localization technology, wireless networks and mobile devices. Additionally, it provides access mechanisms to multiagent system services, through mobile devices, such as mobiles phones or PDA. The ubiquitous access is provided via Wi-Fi wireless networks and the notification and alarm management module is based on SMS and MMS technologies. The user identification and localization system in HoCaMA are based on Java Card and RFID technologies. The system proposed within this work is dynamic, flexible, robust and very adaptable to changes of context. All this makes it an open system, easy to integrate into complex environments that it does not dependent of a specific programming language.

HoCaMA architecture describes four basic blocks that can be seen in Figure 2: Applications, Services, Agents Platform and Communication Protocol. These four blocks constitute the whole functionality of the architecture.

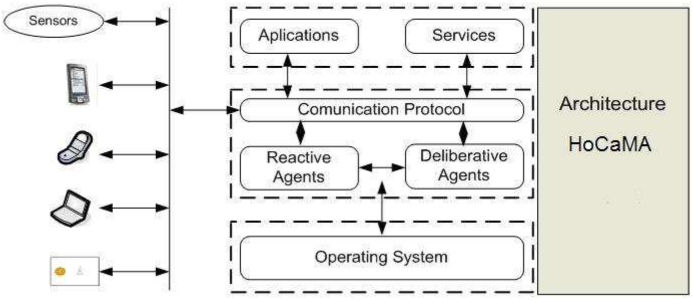


Fig. 2 HoCaMA framework

3.1 Applications

Represent all programs that can be used to exploit the system functionalities. Applications are dynamic and adaptable to context, reacting different according certain situations and the service invoked. They can be executed local or remotely, even on mobile devices with limited processing capabilities because computing tasks are largely delegated to the agents and services.

3.2 Agents Platform in HoCaMA

The Agents 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 3.

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. They 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.

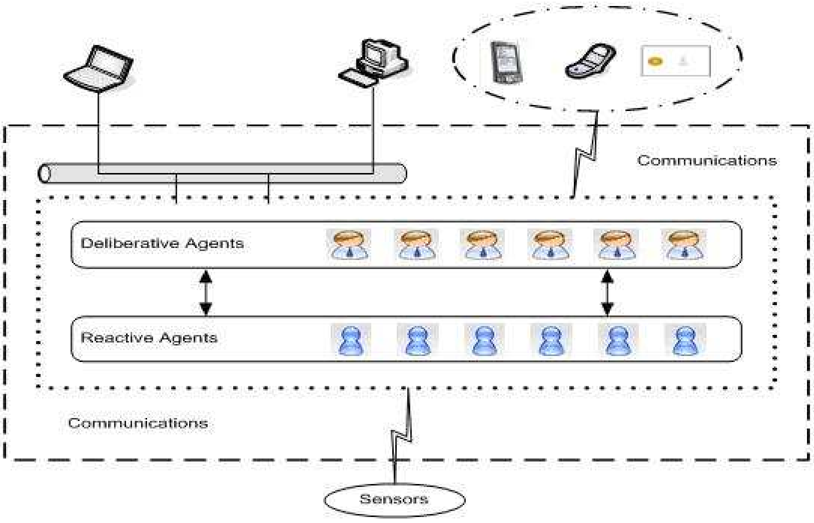


Fig. 3 Agents platform structure in the HoCaMA architecture

The second group is made up of reactive agents. Most of the research conducted within the field of multiagent 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 multiagent 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 or tasks [14]. From this definition, we can define a real-time multiagent system (Real Time MultiAgent System, RT-MAS) as a multiagent 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. In this kind of environments every agent requires autonomy as well as certain cooperation skills to achieve a common goal. In addition, it is necessary to ensure system response in real time, and should be avoided establishing communications between agents too bulky.

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 predefined agents which provide the basic functionalities of the architecture:

3.2.1 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, then the agent creates a new thread to start communication using sockets. The agent sends all requests to the Manager Agent which processes the request. The socket remains open until a response to the specific request is sent back to the application using another XML message. All messages are sent to Security Agent to analyze their structure and syntax.

3.2.2 CoSe Agent

It is responsible for all communications between services and the platform. The functionalities are similar to CommApp Agent but backwards. This agent is always on ”listening mode” waiting for responses of services. Manager Agent indicates CommServ Agent the service that must be invoked. Then, CommServ Agent creates a new thread with its respective socket and sends an XML message to the service. The socket remains open until the service returns a response. All messages are sent to Security Agent to analyze their structure and syntax. This agent also checks periodically the status of all services to know if they are idle, busy, or crashed.

3.2.3 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. The name of the service, description of the service, parameters required, and IP address of the computer where the service is running are some of the information stored in the list of services. However, there is dynamic information that is constantly modified. The service performance (average time to respond requests), the number of executions and the quality of the service. This last data is very important, assigning a value between 0 and 1 to all services. All new services have a quality of service (QoS) value set to 1, this value decreases when the service fails (e.g. service crashes, no service found, etc.) or have a subpar performance compared to similar executions in the past. QoS is increased each time the service processes the tasks assigned efficiently. Information management is especially important on Ambient Intelligence environments because there is very sensitive and personal data processed. Thus, security must be a major concern when developing AmI-based systems, reason why HoCaMA does not implement a service discovery mechanism, so systems must employ only the specified services at a trusted list of services. However, agents can select the most appropriate service (or group of services) to accomplish a specific a task.

3.2.4 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.

3.2.5 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 (CoAp or CoSe) that the message cannot be delivered. This agent also informs the problem to the Directory Agent, which modifies the QoS of the service where message was sent.

3.2.6 Manager Agent

Decides which agent must be called taking into account the QoS and 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. An optimal choice has higher QoS and better performance. Manager Agent has a routing list to manage messages from all applications and services. This agent also checks if services are working properly. It requests the CoSe Agent to send regularly ping messages to each service. If a service does not response, CoSe inform Manager Agent which tries to find an alternate service and informs the Directory Agent to modify the respective QoS.

3.2.7 Interface Agent

This kind of agent has been designed to be embedded in users' applications. Interface agents communicate directly with the agents in HoCaMA 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 rest of the process follows the same guidelines to call any service. Interface agents must be simple enough to allow them execute on mobile devices, such as cell phones or PDA's. All high demanding processes must be delegated to services.

3.3 Services

Represent the activities that the architecture offers. They are the bulk of functionalities of the system at processing, delivery and acquire information level. Services are designed to be invoked locally or remotely. Services can be organized as local services, web services, GRID services, or even as individual stand alone services. Services can make use of other services to provide the functionalities that users require. HoCaMA has a flexible and scalable directory of services, so they can be invoked, modified, added, or eliminated dynamically and on demand. It is absolutely necessary that all services follow the communication protocol to interact with the rest of the architecture components.

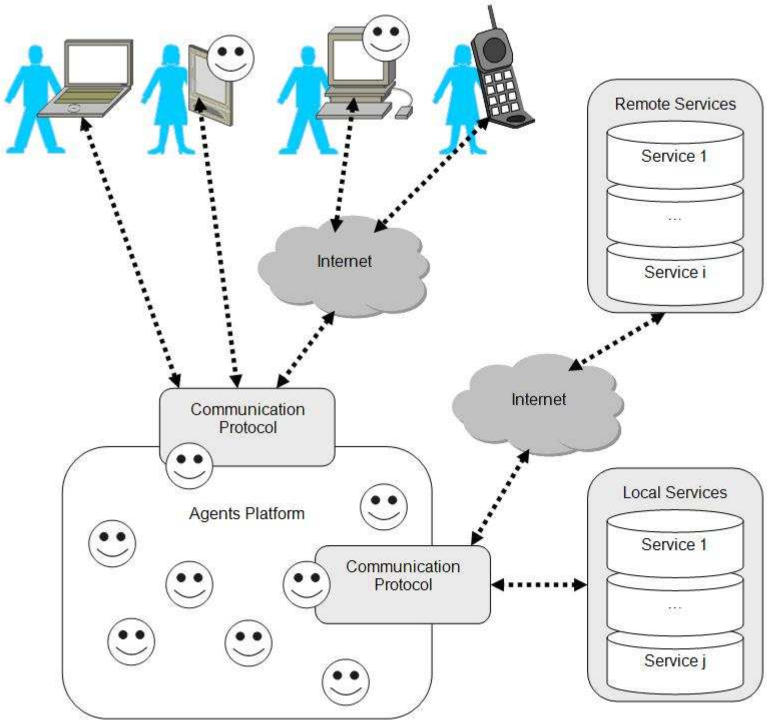


Fig. 4 Request a service by an application

HoCaMA measures and collects data through sensors and mobile devices, as can be seen in Figure 4. A simple case illustrated in Figure 4 can better demonstrate the basic functioning of HoCaMA when it is requesting a service. A user needs to calculate the sum of two numbers and wants to do it through a PDA remotely connected to the system. The user executes a mathematical toolkit which provides him with a large set of formulas from which he selects the sum function, introduces a set of values, and clicks a button to get the result. When the user clicks the button, the application sends a request to the platform to find a service that can process that request. The agents invoke the appropriate service and send the request. The service processes the request and sends the result back to the agents which in turn send it to the application. It is obvious that invoking a remote service to execute a sum is not the best choice. But imagine a large scale process that uses complex AI techniques, such as genetic algorithms, data mining, neural networks, etc. where the limited processing capacity of the PDA makes it impossible to calculate. In this case, the service may be in a powerful computer and could be remotely invoked by the PDA.

3.4 HoCaMA Communication Protocol

Communication protocol allows applications, services and sensors to be connected directly to the platform agents. The protocol presented in this work is open and independent of programming languages. It is based on the SOAP standard and allows messages to be exchanged between applications and services as shown in Figure 5.

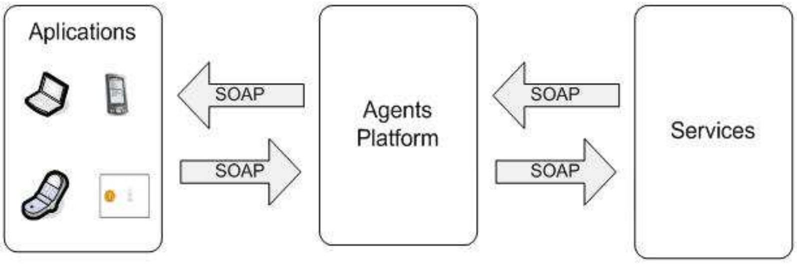


Fig. 5 Communication using SOAP messages in HoCaMA

However, interaction with environmental sensors requires Real-time Transport Protocol (RTP) [13] 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 uncast network services. The RTCP protocol is added to RTP, allowing a scalable form of data supervision. Both RTP and RTCP are designed to work independently from the transport and lower network services. They are in charge of transporting data with real-time characteristics, and of supervising the quality of service, managing the information for all the entities taking part in the current session.

The communications 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.

3.5 Location and Identification System in HoCaMA

The location and Identification system in HoCaMA incorporates Java Card [34] and Radio Frequency Identificacion (RFID) [12], [28] technologies. The primary purpose of the system is to convey the identity of an object or person, as with 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.

3.5.1 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, as shown in Figure 6, 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. As shown in Figure 6 the applet A and B share the same applet context and data, while the applet A shares data with the applet C is at different context. 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) [12] is a wireless communication technology used to identify and receive information on humans, animals or objects in motion [28]. 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 [28]. The transponders are much cheaper and have a size smaller than the chips assets (with batteries), but have a range smaller. 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 be-

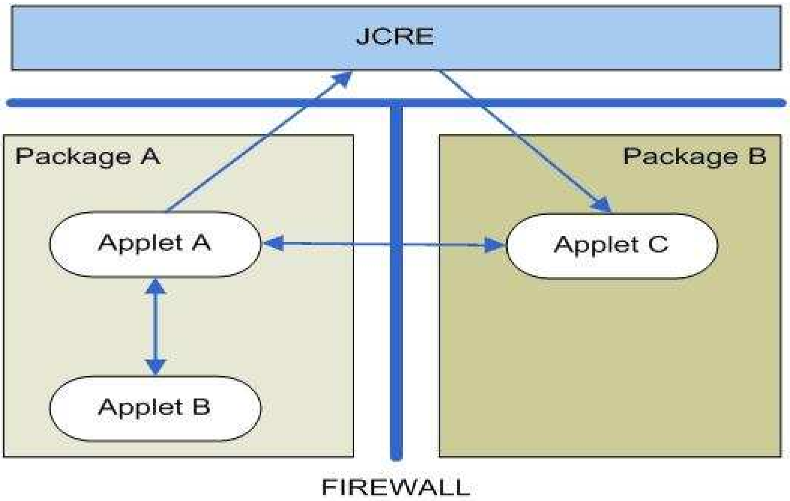


Fig. 6 ”Firewall Model” for security in Java Card

gins 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 [12], [28]. RFID provides more information than other auto-identification technologies, speeds up processes without losing reliability, and requires no human intervention.

The combination of the Java Card and RFID 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. In this sense, RFID is an automated data-capture technology that can be used to electronically identify, track, and store information about patients. It is most frequently used in industrial/manufacturing, transportation, distribution, and warehousing industries, however, there are other growth sectors including health care. HoCaMA uses microchips mounted on bracelets worn on the patient’s wrist or ankle, and sensors installed over protected zones, with an adjustable capture range up to 2 meters. The microchips or transponders help locate the patients, which can be ascertained by consulting the deliberative agents.

3.6 Alert System in HoCaMA

The alert system is integrated into the HoCaMA architecture and uses mobile technology to inform users about alerts, warnings and information specific to the daily routine of the application environment. It gets so improve service quality of communication and control at all times the performance of services that are implemented in architecture.

This is a very configurable system that allows users to select the type of information they are interested, and to receive it immediately on their mobile phone or PDA. It places the information to be sent into information categories. The users determine the information they are interested in. The system automatically sends the information to each of the users as soon as it is available.

3.6.1 Alert System Features

Users wishing to be entered in this service must register, indicating his phone number and email address and giving your permission to send messages to him. It can do so through the application that implements the service validated as users of the same.

Users also from the application, they can select which categories of information you wish to receive messages, so that only receive the information they really useful.

The alert system is proactive, that is, users should not bother to monitor the environment in which the service is implemented, to see if there is information that interests them. It alerts the system itself which cares for users to get information immediately on their mobile devices, with the benefits that this entails, so they can get all this information without having to meet at her workplace. The application also automatically generates alert messages and warnings can also send information on time or scheduled. The application defined a number of operators and reviewers of information that are responsible for inserting and send messages to users who wish to receive such information.

The service alerts and warnings are composed as a sub-application within architecture HoCaMA, but can operate independently of this. The application is implemented with the Struts. It is based pattern Model-View-Controller (MVC). This pattern of development allows us to completely detached three layers that make up the entire web. A model of this type allows the application is fully scalable. As shown in Figure 7 the lines indicate a direct association and association indirect dashed lines.

- Model: In this layer lies the data model of the application. This is communication at all levels with the database.
- View: This is the presentation layer to the user. This is where the data are joined with the design chosen for the user accessing the information. In this layer can be detected that this type of client accessing data to personalize their presentation

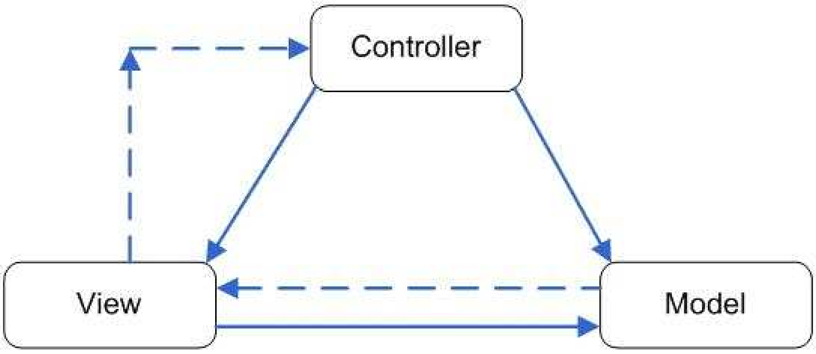


Fig. 7 Relationship between the model/view/controller in the model MVC

giving rise to language versions or versions for different channels such as PDA, SMS, WAP, etc.

- Controller: Here comes the user communications with the application. Every action required by the user is reflected in a flood of calls to data model from the layer of control.

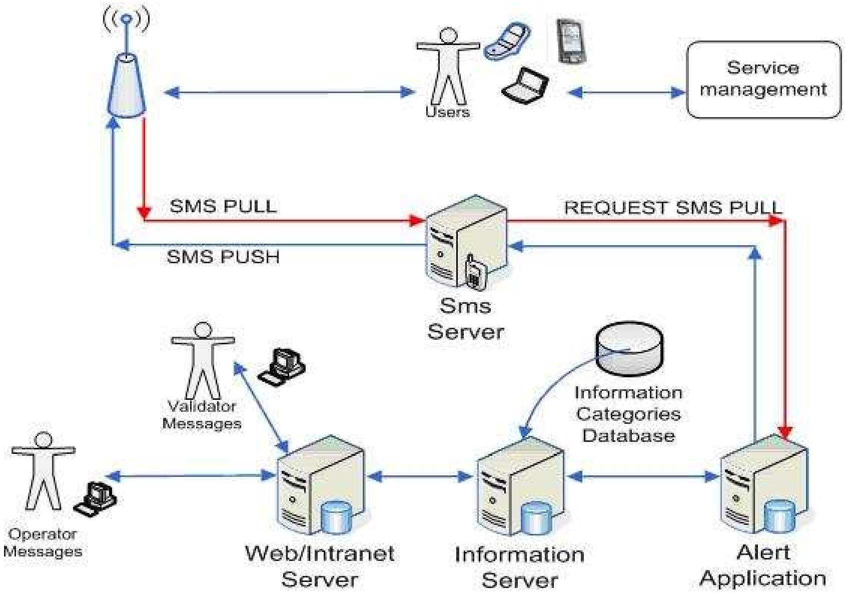


Fig. 8 Alert System framework

Within the application as shown in Figure 8 there are three profiles clearly differentiated, but not exclusive. The application profiles types are: users or subscribers, issuers and reviewers messages. Subscribers have to accept the conditions of the subscription service to introduce or modified their personal data and subscribe to categories wishing. Issuers are responsible for writing messages and confirmation for subsequent validation. Each issuer sends information only to categories assigned it. The reviewers are responsible for validating, if necessary, the messages issued by issuers. The reviewers only reviewed messages of assignment categories.

On the other side as shown in Figure 8 system also handles requests for information by the user via mobile devices, are the called SMS PULL. The messages are sent to a special issue with a specific text, which depends on the information you want to ask, and the system is responsible for managing and providing the information requested. The SMS PUSH is generated by the system automatically without the user's explicit request. In Figure 9 you can see the options available for a reviewer user messages on the alerts system described.

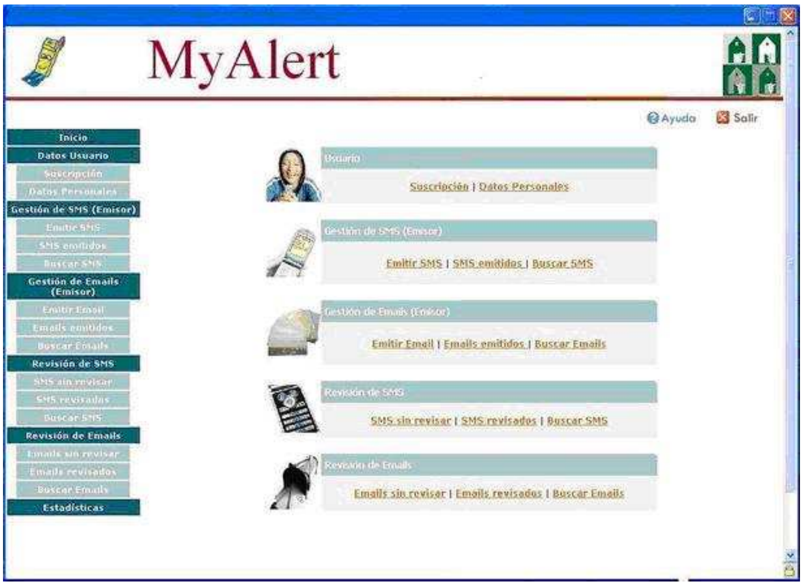


Fig. 9 Alert system implemented in HoCaMA

4 Using HoCaMA to Develope a MultiAgent System for a Home Care Environment

Ambient Intelligence based systems aim to improve people quality of life, offering more efficient and easy to use services and communication tools to interact with other people, systems and environments. One of the most benefited segments of population with the development of Ambient Intelligence based systems is elderly and disabled people.

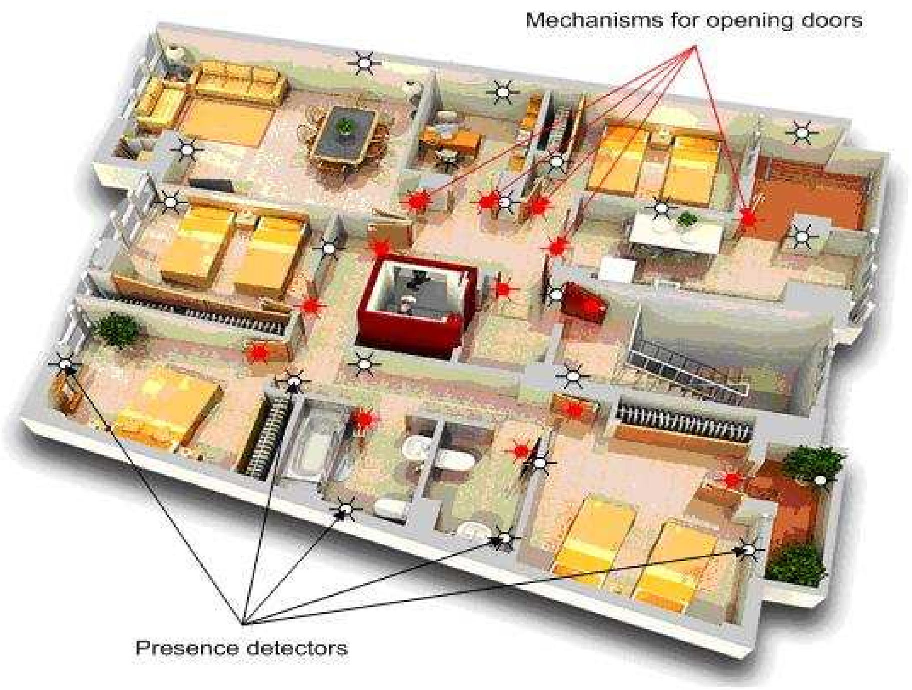


Fig. 10 Home plane

Agents and multiagent systems in environments for disabled people are becoming a reality, especially on health care. Most agents-based applications are related to the use of this technology in patients monitoring, treatment supervision and data mining.

HoCaMA has been employed to develop a multiagent system aimed to enhance assistance and care for low dependence patients at their homes. As shown in Figure 10 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.

Main functionalities in the system include reasoning, planning mechanisms, management alerts and responses in execution time offered to certain stimuli, as shown in Figure 11. 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.

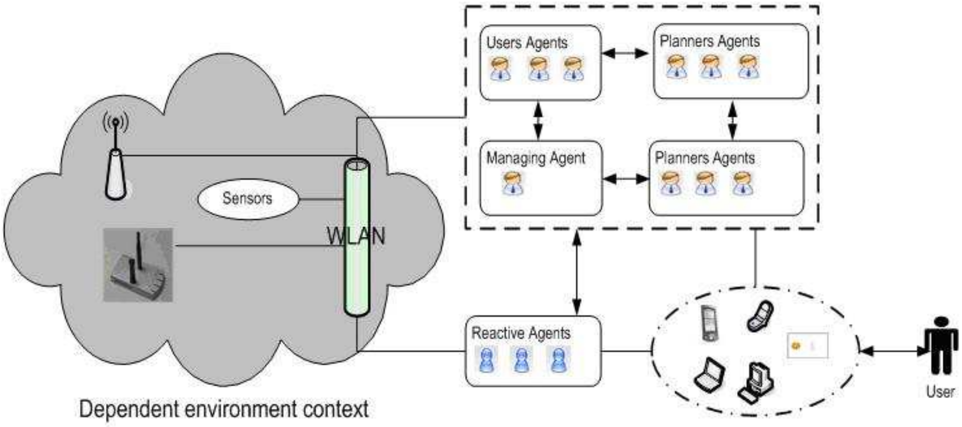


Fig. 11 HoCaMA structure in a home care environment for handicapped persons

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 send. There are priority tasks that a set of agents can perform. This ensures that the priority tasks are always available.

There are four types of new deliberative BDI agents which have been defined in this case study:

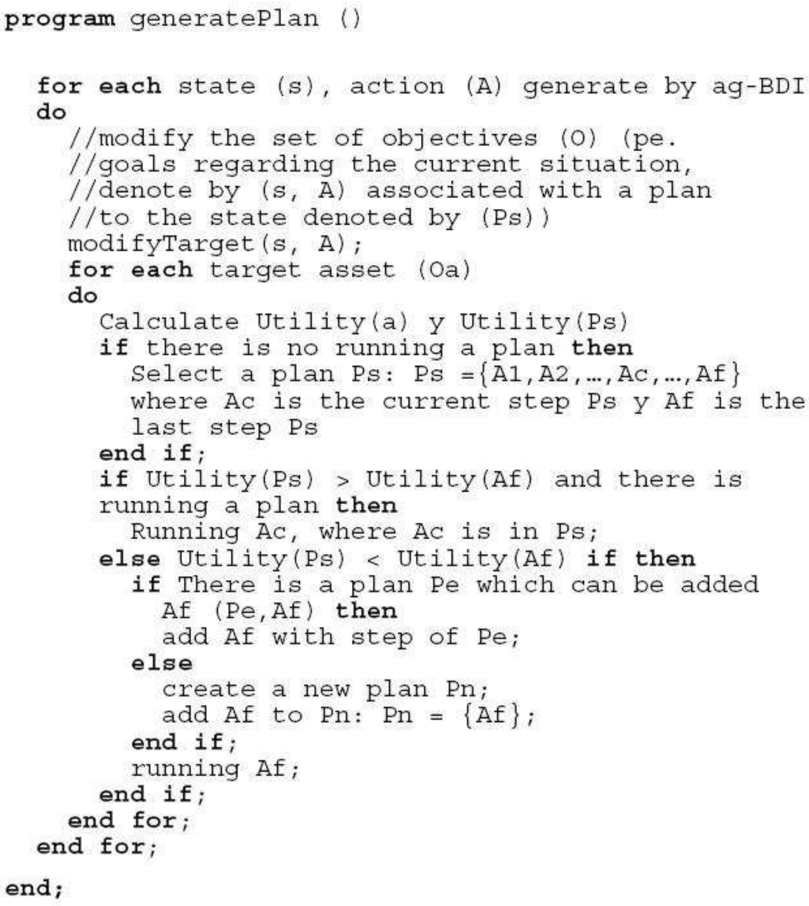
- User Agents. This agent manages the users' personal data and behaviour (monitoring, location, daily tasks, and anomalies). The beliefs and goals used for every user depend on the plan or plans defined by the super-users. User Agent maintains continuous communication with the rest of the system agents, especially with the ScheduleUser Agent (through which the scheduled-users can communicate the result of their assigned tasks) and with the SuperUser Agent. The User Agent must ensure that all the actions indicated by the SuperUser are taken out, sending a copy of its memory base (goals and plans) to the Manager Agent in order to maintain backups. There is one agent for each patient registered in the system.
- SuperUser Agent. It also runs on mobile devices (PDA) and inserts new tasks into the Manager Agent to be processed by a Case-Based 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. There is one agent for each doctor connected to the system.

- ScheduleUser Agent. It is a BDI agent with a Case-Based Planning (CBP) mechanism embedded in its structure. It schedules the users' daily activities obtaining dynamic plans depending on the tasks needed for each user. It manages scheduled-users profiles (preferences, habits, holidays, etc.), tasks, available time and resources. Every agent generates personalized plans depending on the scheduled-user profile. There are as many ScheduleUser Agents as nurses connected to the system. Figure 12 shows the algorithm for plans generation and promotes interaction between actors and sensors. In the algorithm shown in Figure 12, plans are generated when an existing plan is not currently being executed and when actions are executed. When a plan is executing, actions are suppressed and plans are executed to completion. Actions and plans are selected on the basis of respective utility functions. It is part of a state and an action generated by the agent. It changed the objectives and if there is no plan of action in implementing the plan of selecting optimal action. If there is a plan being implemented is associated action if the plan can be added or creating a new action plan to save the platform agents. Finally runs action.
- Manager Agent. It runs on a Workstation and plays two roles: the security role that monitors the users' location and physical building status (temperature, lights, alarms, etc.) trough a continuous communication with the Devices Agent; and the manager role that handle the databases and the tasks assignment. It must provide security for the users and ensure the tasks assignments are efficient. This assignment is carried out through a Case-Based Reasoning (CBR) mechanism, which is incorporated within the Manager Agent. When a new assignation of tasks needs to be carried out, both past experiences and the needs of the current situation are recalled, allocating the respective and adequate task. There is just one Manager Agent running in the system.

On the other hand there are a number of reactive agents that work in collaboration with the deliberative agents. Reactive 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.

In Figure 13 it is possible to see an example corresponding to a warning generated by the system. As can be seen in Figure 13, the system generates the alert after detecting that the user is motionless in the hallway of his home longer than that determine the parameters of a normal situation. The sensors determine all the time the patient's condition at home and communicate this information to a reactive agent, identified as the user agent. The user agent is in charge to send all the sensor information to the central system and determines if it is necessary to generate an alert. When the user agent detects a warning, it automatically communicates the situation to the manager agent. Finally the manager agent, through the SMS alert system available, sends the warning sms message to the health care center and to the patient's family. This warning is recorded in the system for security reasons and

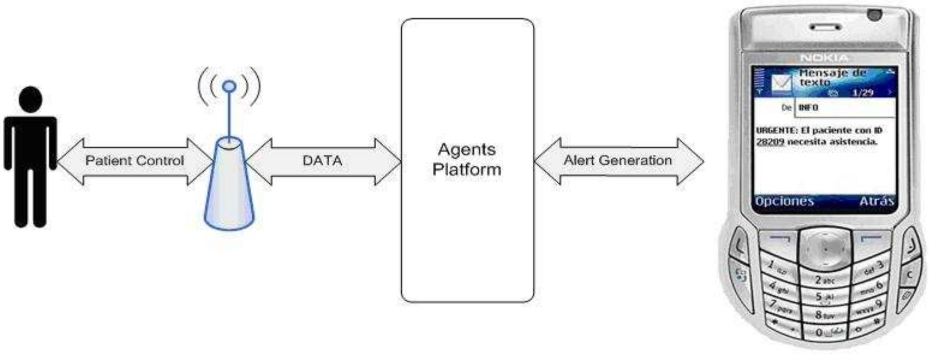


```
program generatePlan ()

  for each state (s), action (A) generate by ag-BDI
  do
    //modify the set of objectives (O) (pe.
    //goals regarding the current situation,
    //denote by (s, A) associated with a plan
    //to the state denoted by (Ps))
    modifyTarget(s, A);
    for each target asset (Oa)
    do
      Calculate Utility(a) y Utility(Ps)
      if there is no running a plan then
        Select a plan Ps: Ps ={A1,A2,…,Ac,…,Af}
        where Ac is the current step Ps y Af is the
        last step Ps
      end if;
      if Utility(Ps) > Utility(Af) and there is
      running a plan then
        Running Ac, where Ac is in Ps;
      else Utility(Ps) < Utility(Af) if then
        if There is a plan Pe which can be added
          Af (Pe,Af) then
          add Af with step of Pe;
        else
          create a new plan Pn;
          add Af to Pn: Pn = {Af};
        end if;
        running Af;
      end if;
    end for;
  end for;

end;
```

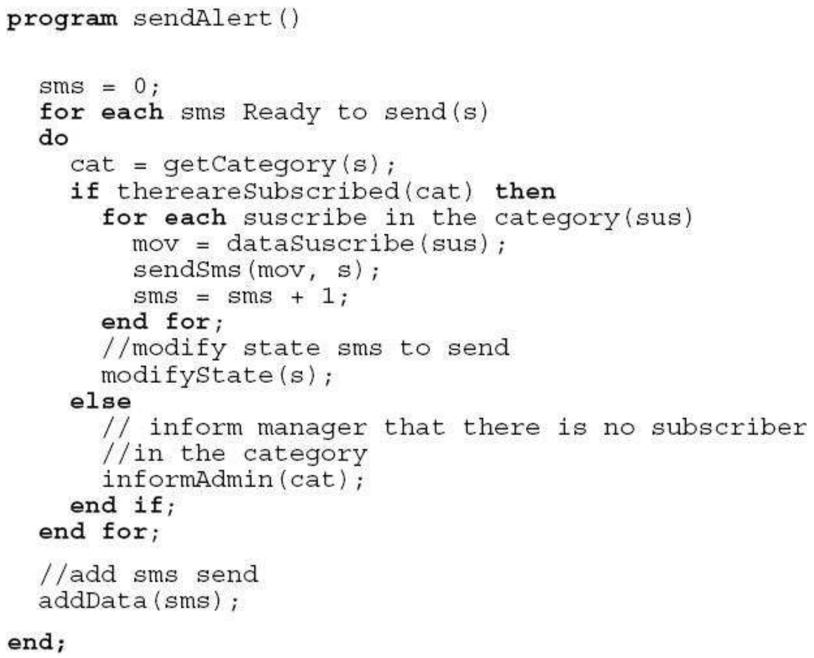
Fig. 12 Algorithm for plan generation in HoCaMA



URGENTE: El paciente con ID 28209 necesita asistencia.

Fig. 13 Alert SMS example sent by the system

for future statistical controls. The steps can be seen in Figure 14 which shows the action of sending a warning by the agent manager.



```
program sendAlert()

  sms = 0;
  for each sms Ready to send(s)
  do
    cat = getCategory(s);
    if thereareSubscribed(cat) then
      for each suscribe in the category(sus)
        mov = dataSuscribe(sus);
        sendSms(mov, s);
        sms = sms + 1;
      end for;
      //modify state sms to send
      modifyState(s);
    else
      // inform manager that there is no subscriber
      //in the category
      informAdmin(cat);
    end if;
  end for;

  //add sms send
  addData(sms);

end;
```

Fig. 14 Algorithm form sending alerts in HoCaMA

5 Results

HoCaMA has been used to develop a multiagent system for monitoring disabled patients at home. The main features of HoCaMA include reasoning and planning mechanisms, and alert and response management. Most of the responses in HoCaMA are reactions in real time to certain stimuli, and represent the abilities that the reactive agents have in the HoCaMA architecture based platform. To offer all these features the system uses various technologies and acquires information from users and the surrounding environment. Some of the technologies used to test the system include mobile technology for managing service alerts through PDAs and mobile phones, and Java Card technology for identification and access control.

One of the main contributions of the HoCaMA architecture is the 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 HoCaMA. The system studies the information collected, and applies a reasoning process which allows alerts to be automatically generated. For the 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 HoCaMA fits perfectly within complex systems by correctly exploiting services and planning mechanisms.

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

Factor	HoCaMA	ALZ-MAS
Average response time to incidents (min.)	8	14
Assisted incidents	12	17
Average number of daily planned tasks	12	10
Average number of daily services completed	46	32
Time employed to attend and alert (min.)	75	90

Table 1 presents the results obtained after comparing the HoCaMA architecture to the previously developed ALZ-MAS architecture [6] in a case study on medical care for patients at home. The ALZ-MAS architecture allows the monitoring of patients in geriatric residences, but home care is carried out through traditional methods. The case study presented in this work consisted of analysing the functioning of both architectures in a test environment. The HoCaMA architecture was implemented in the home of 5 patients and was tested for 30 days. The results were very 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 handicapped care services providers, whose work performance improved, allowing them to avoid unnecessary movement such as travels and visits simply oriented to control or supervise the patient. 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 HoCaMA architecture since patients can live in their own homes with the same level of care as those offered at the residence.

6 Conclusions

The growing numbers of elderly population impose an urgent need to develop new approaches to care provision. The convergence of a number of technologies such as multiagent systems, federated information management, safe communications, hypermedia interfaces, rich sensorial environments, increased intelligence of home appliances, and collaborative virtual environments, represents an important enabling factor for the design and development of virtual elderly support community environments. In particular, a platform based on agents combined with federated information management mechanisms provides a flexible infrastructure on top of which specialized care services can be built. Participants in this community include the elderly people, their relatives, and care center personnel as shown in Figure 15. We chose to tackle the home-monitoring issue in a more global way rather than in an only individual-centred way. This collective vision makes it possible to release individuals' patterns who will allow the system answering current health problems. This large-scale and global solution requires setting up a strongly distributed and dynamic system. Because classical classification methods are not adapted to this context, we had to propose a new distributed classification method.

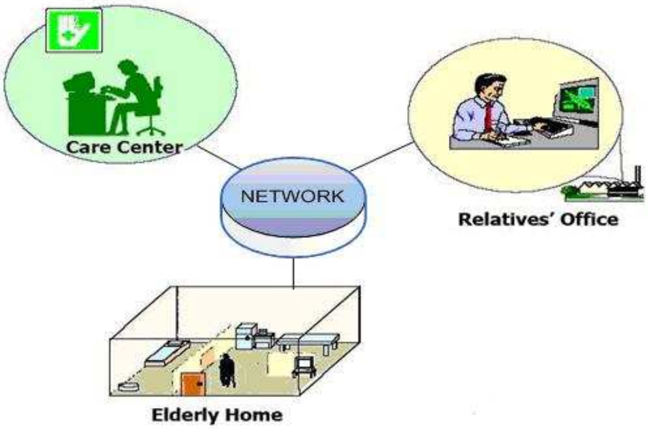


Fig. 15 Actors' communications in HoCaMA

The HoCaMA architecture presented in this paper is a step ahead in the development of Ambient Intelligence environments for Home Care. The architecture proposes novel systems for alert management and patient location and identification. However there is still to much work to be done, for example in patients attention and safety. That is our next step.

Acknowledgements This work has been partially supported by the Spanish Ministry of Science and Technology project TIN2006-14630-C03-03, the UPSA project U05EIA-07L01 and the Spanish Ministry of Labor and Social Security project "Plataforma inteligente para la gestión integral de residencia geriátricas".

References

1. Anastasopoulos M., Niebuhr D., Bartelt C., Koch J. and Rausch A.: Towards a Reference Middleware Architecture for Ambient Intelligence Systems. In: ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (2005)
2. Angulo C. and Tellez R.: Distributed Intelligence for smart home appliances. In: Tendencias de la minería de datos en Espaa. Red Española de Minería de Datos. Giráldez R., Riquelme J.C. and Aguilar-Ruiz J.S., pp. 1-12 (2004)
3. Ardissono L., Petrone G. and Segnan M.: A conversational approach to the interaction with Web Services. Computational Intelligence, Blackwell Publishing, **20**, 693–709.(2004)
4. Bahadori S., Cesta A., Grisetti G., Iocchi L., Leone R., Nardi D., Oddi A., Pecora F. and Rasconi R.: . RoboCare: Pervasive Intelligence for the Domestic Care of the Elderly. Artificial Intelligence. **1(1)**, 16–21 (2003)
5. Bonino da Silva L.O., Ramparany F., Dockhorn P., Vink P., Etter R. and Broens T.: A Service Architecture for Context Awareness and Reaction Provisioning. In: IEEE Congress on Services, pp. 25–32 (2007)
6. Carrascosa C., Bajo J., Julián V., Corchado J.M., Botti V.: Hybrid multi-agent architecture as a real-time problem-solving model. Expert Systems With Applications. **34(1)**, 2–17. (2008)
7. Cesta A., Bahadori S., Cortellesa G., Grisetti G., Giuliani, M., et al.: The RoboCare Project, Cognitive Systems for the Care of the Elderly. In: Proceedings of International Conference on Aging, Disability and Independence (ICADI), Washington D.C., USA (2003)
8. Chan M., Campo E. and Est'eve D.: PROSAFE, a multisensory remote monitoring system for the elderly or the handicapped. In: Proceedings of the 1st International Conference on Smart Homes and Health Telematics (ICOST '03), Paris, France, pp. 89–95 (2003)
9. Corchado J.M., Bajo J., de Paz Y., Tapia D.: Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care. Decision Support Systems, **34(2)**, 382–396 (2008).
10. Corchado J.M., Bajo J. and Abraham A.: GERAmI: Improving the delivery of health care. IEEE Intelligent Systems. **23(2)**, 19–25 (2008)
11. IST Advisory Group: Ambient Intelligence: From vision to reality. European Commission (2003)
12. ITAA. (2004). Radio Frequency Identification. RFID...coming of age. In White paper, Information Technology Association of America. http://www.itaa.org/rfid/docs/rfid.pdf.
13. Jacobsen V., Fredrick R., Casner S., and Schulzrinne H.: RTP: A transport Protocol for Real-Time Applications. In: RFC 1889. Lawrence Berkeley National Laboratory, Xerox PARC, Precept Software Inc., GMD Fokus, January 1996. Online. Internet. Available via DIALOG. http://www.connect.org.uk./teckwatch/cgi-bin/rfcshow?1889
14. Julián V. and Botti V.: Developing real-time multi-agent systems. Integrated Computer-Aided Engineering. **11(2)**, 135–149 (2004)
15. Kleindienst J., Macek T., Seredi L. and Sedivy J.: Vision-enhanced multi-modal interactions in domotic environments. In: 8th ERCIM workshop "User interfaces for all" (2004) Available via DIALOG.
http://www.ui4all.gr/workshop2004/
16. Li Y., Shen W. and Ghenniwa H.: Agent-Based Web Services Framework and Development Environment. Computational Intelligence. **20(4)**, 678–692 (2004)
17. Liu, X.: A Multi-Agent-Based Service-Oriented Architecture for Inter-Enterprise Cooperation System. In: Proceedings of the Second international Conference on Digital Telecommunications (ICDT'07). IEEE Computer Society, Washington DC (2007)
18. Mengual L., Bobadilla J. and Triviño G. (2004). A fuzzy multi-agent system for secure remote control of a mobile guard robot. In: Second International Atlantic Web Intelligence Conference, AWIC 2004, LNCS 3034, pp. 44–53.
19. Moreno A. and Garbay C.: Special issue on Software Agents in Health Care. Artificial Intelligence in Medicine. **27** (2003)

20. Noury N., Villemazet C., Barralon P. and Rumeau P.: Ambient multi-perceptive system for residential health monitoring based on electronic mailings Experimentation within the AILISA project. In: Proceedings of the 8th IEEE International Conference on e-Health Net-working, Applications and Services (HEALTHCOM '06),New Delhi, India, pp. 95-100 (2006)
21. Ricci A., Buda C. and Zaghini N.: An agent-oriented programming model for SOA and web services. In: 5th IEEE International Conference on Industrial Informatics, pp. 1059-1064 (2007)
22. Richter K. and Hellenschmidt M.: Interacting with the Ambience: Multimodal Interaction and Ambient Intelligence. In: Position Paper to the W3C Workshop on Multimodal Interaction, pp. 19-20 (2004)
23. Rigole P., Holvoet T., Berbers Y.: Using Jini to integrate home automation in a distributed software-system. In: 4th International workshop on distributed communities in the Web, LNCS 2468, pp. 291–304(2002)
24. Sakka E., Prentza A., Lamprinos I. E., Leondaridis L. and Koutsouris D.: Integration of monitoring devices in the e-Vital service. In: Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEMBS '04), vol. 4, pp. 3097-3100, San Francisco, Calif, USA (2004)
25. Shafiq M.O., Ding Y. and Fensel D.: Bridging Multi Agent Systems and Web Services: towards interoperability between Software Agents and Semantic Web Services. In: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06). IEEE Computer Society, Washington, DC. pp. 85-96. (2006)
26. Tafazolli R. and Saarnio J.: eMobility : Mobile and Wireless Communications Tecnology Platform. Strategic Research Agenda. Versin 4 (2005)
27. Tapia D.I., Bajo J., De Paz F., Corchado J.M.: Hybrid Multiagent System for Alzheimer Health Care. In: Hybrid Artificial Intelligence Systems. HAIS 2006. Javier Bajo, Emilio S. Corchado, lvaro Herrero and Juan M. Corchado (Eds.). Universidad de Salamanca, pp. 1–18 (2006)
28. U.S. Department of Commerce: Radio Frequency Identification. Opportunities and Challenges in Implementation. (2005) Available via DIALOG.
http://www.technology.gov/reports/2005/
29. Walton, C.: Agency and the Semantic Web. Oxford University Press, Inc.(2006)
30. Want R., Pering T., Borriello G. and Farkas K.I.: Disapearing Hardware. Pervasive Computing. **1(1)** (2002)
31. Want R.: You Are Your Cell Phone. IEEE Pervasive Computing. **7(2)**, 2–4 (2008)
32. Weiser M.: The Computer for the Twenty-First Century. Scientific American. **265**, 94–104 (1991)
33. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley and Sons, Chichester, England.(2002)
34. Zhiqun C.: Java Card Technology for Smart Cards. Prentice Hall, Boston (2000)