UBIMATE: A ubiquitous healthcare emergency platform

Dimitrios Al. Alexandrou, Panagiotis Ag. Gouvas, and Athanasios D. Bouras

***Abstract* — This paper presents a multi-agent platform for the confrontation of emergency incidents of patients that belong to High Risk Groups. It presents the design and implementation of the required software components. Moreover, the reader is presented with all the required information about the discrete components and their communication so as to provide the desired service. Finally, the paper presents the actions that will be performed in the future concerning its further enhancement.**

I. Introduction

At present, the majority of people around the world that suffer from chronic diseases or belong to high risk groups (HRG) are unwilling to venture out of the confines of their homes without the escort of dedicated people, lest they may require emergency treatment. Consequently, many of them progressively become socially disabled, which aggravates their already delicate physical condition. The proposed system will try to address the aforementioned problem by offering the end user/patient an increased degree of self-sufficiency when going out of his/her home. For this purpose, it introduces an intelligent rescue mechanism that provides instant and efficient assistance in the event of an emergency. Hence, the patient benefits from an improved level of safety; stress is reduced and quality of life is ameliorated.

More specifically, our team designed and implemented a distributed intelligent software platform which performs a set of actions when a medical emergency occurs. First of all, it undertakes the task to contact the nearest available and suitable doctor and direct him/her to the patient's exact location. Furthermore, it alerts the nearest available hospital so that an ambulance may be dispatched to the site of the emergency. Therefore, the patient receives help from both the doctor and the hospital that are closest to his/her location.

The platform that was designed and developed is called UBIMATE, which stands for Ubiquitous Healthcare MATE. UBIMATE proposes a solution based on mobile devices and geographical information so as to provide immediate medical assistance to chronically ill patients. The patient will use a Personal Digital Assistant (PDA) equipped with a GPS receiver [4], [5], [11], a GPRS modem and "One - Button" functionality in order to utilize the service provided by the platform.

II. System Requirements

A. Introduction

The main objective of UBIMATE system is to provide instant and efficient assistance to a chronically ill patient and save him/her in an emergency. Our aim is to meet this objective as strictly as possible. The patient is the primary end user of the proposed architecture. Furthermore, the secondary user is the doctor who takes the responsibility to provide first aid to the patient. Finally, the hospital and the ambulance crew constitute the ternary user of the system. Our main concern was also to take under consideration the high risk groups (HRGs) and taxonomies of the patients, as well as the specialization of the doctors. In order to obtain the necessary information and knowledge we encountered an iterative procedure of meetings with medical staff that provided us with crucial information concerning the patients and the corresponding doctors. This information also helped us in taking major decisions concerning decision algorithms about the doctor and hospital selection in case of emergency. All the information obtained led us to the creation of the Use Cases below which show the interaction of each of the entities with the UBIMATE architecture.

B. Patient Use Case

The patient is the main actor of the implemented system. The patient performs clearly defined interactions with the system so as to receive the help s/he needs at the time of a medical emergency. The first action performed is the Login operation. Once the patient turns on his/her mobile device, s/he is requested to log in to the system. The Login operation is required so as to begin to be identifiable by UBIMATE platform.

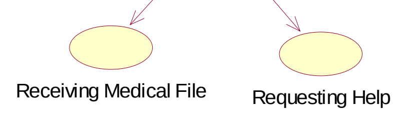

Fig. 1 - Patient Use Case

Manuscript received on June 30, 2006. The UBIMATE platform was developed in the frames of a scientific contest between Universities.

D. A. Alexandrou is a Research Associate in the National Technical University of Athens, Athens, Greece. He is in the Department of Electrical and Computer Engineering (telephone: +30 210 772 2895; fax: +30 210 772 2550; e-mail: dalexan@mail.ntua.gr).

P. A. Gouvas is a Research Associate in the National Technical University of Athens, Athens, Greece. He is in the Department of Electrical and Computer Engineering (e-mail: pgouvas@mail.ntua.gr).

A. D. Bouras is a Research Associate in the National Technical University of Athens, Athens, Greece. He is in the Department of Electrical and Computer Engineering (e-mail: bouras@mail.ntua.gr).

Furthermore, in case of emergency the patient requests help instantly, by pressing a dedicated button on his/her mobile terminal. Once the system receives the alert, performs all the necessary operations so as to provide help to the patient as soon as possible. It is also sending the patient's medical file in a secure way to his/her mobile terminal in order to provide vital information to the doctor/rescuer that will reach the point of the incident.

C. *Doctor Use Case*

The doctor is the second entity of UBIMATE architecture. The doctor begins his/her interaction with UBIMATE platform by logging in.

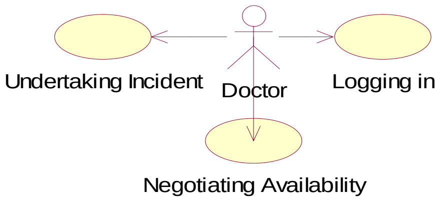

Fig. 2 - Doctor Use Case

The login procedure provides the system with the ability to be informed about the number and identity of the doctors that are online and can accept alerts for emergencies. Moreover, when an incident occurs several online doctors receive an alert. At that moment, a negotiation of availability begins. The doctor replies whether s/he is available or not, using a simple procedure. After collecting responses from doctors, the system executes an algorithm and selects the best available doctor. Finally, the selected one receives a confirmation message from the system and a map showing the patient's exact location along with his/her own.

D. *UBIMATE System Use Case*

Finally, the third entity of the platform is the system itself. As presented in the next figure, the system initially receives an alert for an emergency occurrence. The message is processed immediately and the patient taxonomy and location are retrieved. Hence, the negotiation for a doctor begins by sending messages to the "online" doctors. Concurrently, UBIMATE platform performs a search for the nearest available hospital, as well as retrieving the medical file of the patient. Finally, the system attaches the best available doctor to the incident and sends him/her the map presenting the exact positions of both the patient and the doctor himself. At the same time, the patient's medical file is downloaded to his PDA so as to provide vital medical information concerning his medical profile.

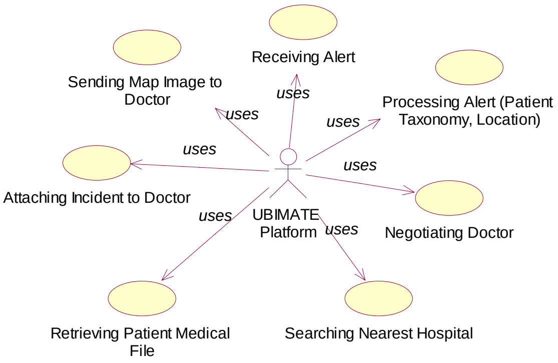
Receiving Alert
Sending Map Image to Doctor
uses
uses
Processing Alert (Patient Taxonomy, Location)
uses
uses
uses
Attaching Incident to Doctor
uses
UBIMATE Platform
uses
Negotiating Doctor
Retrieving Patient Medical File
Searching Nearest Hospital

Fig. 3 - UBIMATE Use Case

III. System Architecture

The system architecture is presented in Fig. 4. Three main domains can be distinguished. The first domain of our architecture is the UBIMATE Agent Container [7]. The Agent Container comprises the agents responsible for receiving the alert from the patient, the negotiating agent that will be responsible for finding an available doctor nearby and the alert forwarder agent that will find the nearest hospital and will forward the alert to it so as to send an ambulance to the location of the emergency as soon as possible. The Agent Container uses two databases. The first one constitutes a map repository that holds the maps needed in order to inform the doctor about the exact position of the patient. The other database stores information about the patients (username, password, medical record etc.), the doctors (available doctors, specialization, etc.) and the hospitals (availability, position, clinics, etc.). All this information is used by the agents so as to decide which doctor to bind to the incident and which hospital to inform about it.

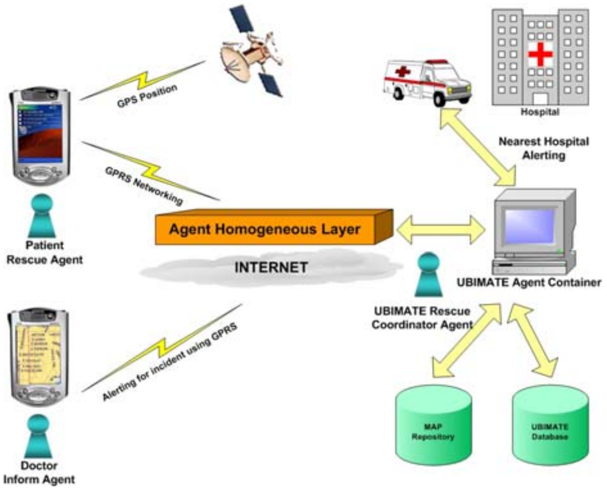

Fig. 4 - UBIMATE System Architecture

The other domain of the above architecture is the patient's domain. The patient possesses a PDA with GPS and GPRS functionality. The PDA also contains a lightweight agent container that holds an alert agent. Using "One - Button" functionality, the patient initializes the agent when he does not feel well. At that moment, the agent begins communicating with the Main Container by sending the needed data for processing. Finally, the third domain is that of the “mobile” doctor. The doctor has a PDA or a laptop. As soon as the Main Container processes the alert from the patient, it establishes communication with available doctors around the area of the emergency. Consequently, a doctor is bound to the incident, receives a map with the exact location of the patient and immediately tries to reach him.

IV. UBIMATE Agent Architecture

The aforementioned Use Cases and User Requirements were transformed into an intelligent multi-agent architecture [6], [9]. The latter, which is depicted below, in Fig. 5, constitutes the UBIMATE intelligent platform and is responsible for providing the functionality as this was extracted from the Use Cases.

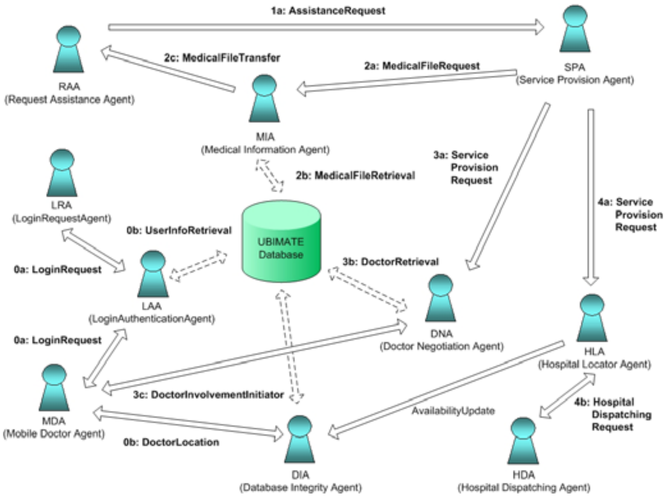

Fig. 5 - UBIMATE Agent Architecture

As shown above, the agent infrastructure contains agents that are installed in different nodes of hardware. The LoginRequestAgent (LRA) and RequestAssistanceAgent (RAA) are installed in the patient’s mobile device [8], [14]. Additionally, the MobileDoctorAgent (MDA) is installed in the doctor’s personal digital assistant (PDA). On top of that, ServiceProvisionAgent (SPA), MedicalInformationAgent (MIA), DoctorNegotiationAgent (DNA), DatabaseIntegrityAgent (DIA) and HospitalLocatorAgent (HLA) are installed in the main server (agent container) of the implemented software platform. Moreover, the HospitalDispatchingAgent (HDA) is located in the premises of each hospital that subscribes to the UBIMATE system. Finally, another crucial component of our infrastructure is the UBIMATE Database that contains vital information about patients, doctors and hospitals.

In the following section we present the functionality of each of the above presented agents.

a. *Login Request Agent (LRA)*

Login Request Agent is the agent that is responsible for identifying the user/patient of UBIMATE and personalizing his information. This agent is initialized once the patient turns on his mobile device and wishes to log in to the system. The agent prompts him to provide username and password parameters so as to be under the attention of UBIMATE.

b. Request Assistance Agent (RAA)

Request Assistance Agent is the agent that is responsible for triggering the main functionality features of UBIMATE. This agent is initialized after the log in procedure of the patient. RAA is polling the GPS receiver continuously so as to acquire valid positional data. Once an emergency occurs, the agent retrieves the ID of the patient and his exact geographic position and sends an alert message to the UBIMATE platform.

c. Mobile Doctor Agent (MDA)

Mobile Doctor Agent is the agent that is responsible for both identifying the doctor and making him available for receiving alert messages from UBIMATE. This agent is initialized when the doctor powers on his mobile terminal. The agent prompts him to provide username and password parameters so that the UBIMATE system is aware of his availability. Furthermore, MDA interacts with UBIMATE in the case of an emergency by sending a confirmation message about whether the doctor is able to offer assistance. Last but not least, it is responsible for updating the doctor’s position.

d. Login Authentication Agent (LAA)

Login Authentication Agent is the agent that is responsible for authenticating both users/patients and doctors subscribing to the UBIMATE platform. This agent receives a message once the patient/doctor turns on his mobile device and wishes to log in to the system. The agent receives the username/password parameters and performs a database search so as to authenticate the person making the request. Once the person is authenticated, LAA sends an acknowledgement message to his device.

e. Service Provision Agent (SPA)

Service Provision Agent is one of the main agents in the UBIMATE platform. SPA receives a message from the patient’s mobile terminal when an emergency takes place. Instantly, SPA communicates concurrently with MIA, DNA and HLA so as to inform them about the incident and sends them the appropriate information in order to enable them to accomplish specific tasks for the patient’s rescue [13]. Afterwards, SPA waits for acknowledgements from the other agents in order to coordinate the rescue procedure. Finally, SPA clones himself so as to create another SPA ready to accept new alerts.

f. Medical Information Agent (MIA)

Medical Information Agent is the agent that is responsible for retrieving the patient's medical file from the UBIMATE database. This agent receives a message from SPA in case of emergency. The agent receives the patientID parameter and performs a database search so as to send a summary of the corresponding medical file to the patient's mobile terminal, taking under consideration the secure delivery of sensitive personal data [12].

g. Doctor Negotiation Agent (DNA)

Doctor Negotiation Agent [10] is the agent that undertakes the task to find a suitable doctor and direct him to the site of the emergency. First of all, DNA searches the UBIMATE database for doctors who are registered as available and in the vicinity of the incident. Afterwards, it contacts all the above doctors concurrently via their respective MDAs and awaits confirmation of availability from each one. Based on their replies, the doctor who is most capable of assisting the patient is chosen. DNA then sends a dispatch confirmation message to the selected doctor, which includes a map with the patient's location along with his own. The doctor selection is performed using an ***algorithm*** which shall be described in detail.

- Firstly, the UBIMATE database search returns all doctors who are logged in the system and are within a 1000-meter radius from the patient, ordered by distance. The smallest of these distances shall be called the *ideal distance*.

- After that, doctors are contacted and the *negotiation phase* begins. This phase ends when ***any*** of the following three conditions is met:
  - a 25-second timeout expires;
  - all contacted doctors have replied, either affirmatively or not;
  - a doctor replies affirmatively and his distance from the patient does not exceed the ideal distance by more than 100 meters.

- Following the aforementioned phase, DNA proceeds to make its selection.

Among the doctors that have replied affirmatively, DNA chooses the one who is nearest the patient, assuming that he will be able to reach the site of the emergency sooner than the others. However, doctors whose specialization does ***not*** match the patient's taxonomy are subjected to a *handicap*. The latter consists of adding 500 meters to their real distance from the patient, for the purposes of the selection. It is noteworthy that this handicap also precludes such doctors from satisfying condition (c) above. Hence, DNA shows a strong preference for doctors who are most suitably trained and experienced to treat the patient in question.

h. Hospital Locator Agent (HLA)

Hospital Locator Agent is the agent that is responsible for discovering the available hospital that is nearest to the site of the emergency. HLA sends a message to the selected hospital's HDA providing the exact location of the incident so as to dispatch an ambulance crew to rescue the patient.

i. Database Integrity Agent (DIA)

Database Integrity Agent constitutes a general-purpose agent, which is responsible for the continuous update of the database. Its main duty is to maintain the database integrity concerning the location of the doctors and the availability of the hospitals. Consequently, makes the service the UBIMATE platform provides safer and more efficient, since it is based on reliable information and data.

j. Hospital Dispatching Agent (HDA)

Hospital Dispatching Agent is the agent that is responsible for accepting rescue alerts from UBIMATE. HDA is located in the hospital premises and receives a message from HLA (Hospital Locator Agent) when an emergency occurs. The agent receives a message from HLA with the map containing the exact position of the incident. Furthermore, it is responsible for sending an update message to DIA concerning the availability status of the specific hospital.

V. System Functionality

UBIMATE architecture as described above comprises a multi-agent system (MAS). In order to summarize the agent presentation, a State-chart diagram is depicted below. The specific diagram shows the distinct states that the system traverses and the actions that trigger these devolvements.

As shown in the Fig. 6, the system operates when an emergency occurs and the patient requests help. The RAA (Request Assistance Agent) sends an Assistance Request message from patient's Mobile Device. SPA (Service Provision Agent) receives the message and initializes three parallel flows of messages so as to offer help to the patient as quickly as possible. The first flow concerns the retrieval of patient's medical information. MIA (Medical Information Agent) receives a message from SPA containing the patient's ID. Hence, it performs a database search in UBIMATE database so as to retrieve the corresponding medical file. Once retrieved, MIA forwards the Medical File to patient's Mobile Device. The information is downloaded and presented on the PDA screen to provide further information to the doctor.

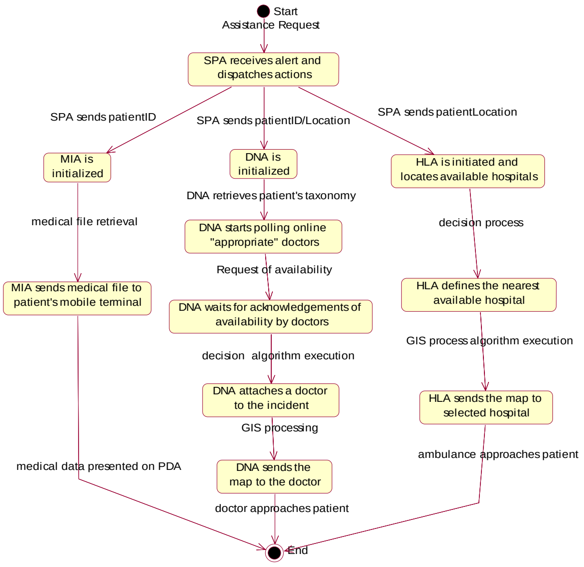
Start
Assistance Request
SPA receives alert and dispatches actions
SPA sends patientID
SPA sends patientID/Location
SPA sends patientLocation
MIA is initialized
DNA is initialized
HLA is initiated and locates available hospitals
DNA retrieves patient's taxonomy
medical file retrieval
DNA starts polling online "appropriate" doctors
decision process
Request of availability
MIA sends medical file to patient's mobile terminal
DNA waits for acknowledgements of availability by doctors
HLA defines the nearest available hospital
GIS process algorithm execution
decision algorithm execution
DNA attaches a doctor to the incident
HLA sends the map to selected hospital
GIS processing
ambulance approaches patient
medical data presented on PDA
DNA sends the map to the doctor
doctor approaches patient
End

Fig. 6 - UBIMATE Functional Diagram

Moreover, the second main flow of messages deals with the negotiation process for the selection of an available and suitable doctor. DNA (Doctor Negotiation Agent) receives the alert from SPA and initializes the procedure of Doctor Negotiation. Firstly, the patient's taxonomy is retrieved using the patient's ID. Once the patient's taxonomy is retrieved, DNA begins polling for available doctors. After the acknowledgements' reception, the decision algorithm is executed so as to choose the most appropriate available doctor. The algorithm contains constraints regarding the doctor's distance from the incident and his specialization depending on user's taxonomy. Once selected, the doctor receives an acceptance message from DNA as well as the map that shows both his and the patient's positions. Hence, the doctor is ready to promptly approach quickly the site of the emergency.

Finally, the third main flow of messages is responsible for the selection of the nearest available hospital. HLA (Hospital Locator Agent) receives the alert message from SPA and initiates the procedure of hospital allocation. HLA performs a database search so as to retrieve the available hospitals. Afterwards, it executes an algorithm of selection taking under consideration the distance of each hospital from the site of the incident. Once the hospital is selected, HDA (Hospital Dispatching Agent) receives a message containing the map that depicts both the patient's and the hospital's positions. The map is forwarded to an ambulance crew which begins to approach the site of the emergency. Consequently, the patient receives help from both the doctor and the hospital as quickly as possible.

VI. INTRA-COMMUNICATION ISSUES

One of the main concerns during the implementation phase was the intra – communication messages between the intelligent agents. The agents constituting the UBIMATE architecture need a common standard so as to communicate with each other[1]. The system's architecture introduced some constraints that had to be taken under consideration during the System Specification phase. Hence, it was decided to use an Ontology as a common vocabulary in order to enhance the communication between the agents [2].

The Artificial-Intelligence literature contains many definitions of an ontology; many of these contradict one another. In general, an ontology is a formal explicit description of concepts in a domain of discourse (classes (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (slots), and restrictions on slots (facets) [3]. An ontology combined with a set of individual instances of classes constitutes a knowledge base. In reality, there is a fine line where the ontology ends and the knowledge base begins. The main reasons that led to the usage of such a semantic representation of information are the following:

- To share common understanding of the structure of information among software agents regarding data exchange.
- To enable reuse of semantically structured data so as to establish a meta-layer of information and further support the expandability of UBIMATE platform.
- To make domain assumptions explicit and consequently accommodate the development phase.

Hence, during the development procedure of the UBIMATE Ontology the following tasks were accomplished:

- defining classes in the ontology;
- arranging the classes in a taxonomic hierarchy;
- defining slots and describing allowed values for these slots;
- filling in the values for slots for instances;

As a result, the Ontology Schema as this was designed and implemented is depicted below:

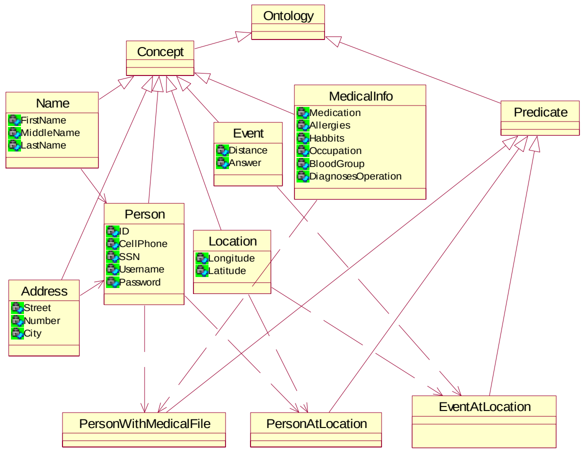

Fig. 7 - UBIMATE Ontology Class Diagram

VII. FUTURE WORK

The main issue that could be further studied and optimized is the negotiation procedure. The negotiation issue regarding the selection of the most appropriate doctor can be further enhanced both concerning the protocol and the implementation. One main enhancement in the specific procedure could be the design and creation of a Negotiation Ontology that would be responsible for the data exchange between the corresponding agents.

VIII. Conclusion

Our focus was on developing an intelligent mobile platform so as to provide assistance to chronically ill people and to High Risk Groups (HRGs). The main concern was to enable physically sensitive groups of individuals to move safely out of the confines of their homes without requiring constant attention by dedicated people. It is firmly believed that such a service will give them the opportunity to develop a richer social life, relieve them from a source of stress and, most importantly, it may save their life in an emergency.

Consequently, recent advancements in the fields of mobile communications and global positioning were closely studied. Then, a way to harness them for the above-mentioned purpose was devised, using state-of-the-art software technologies to ensure seamless cooperation between hardware components.

Such a system ought to be distributed, by its nature. Its implementation was feasible because of the usage of intelligent agents. The autonomy of each intelligent component enabled the implementation of distinct parts of the platform following a distributed and concurrent procedure. Furthermore, integration was achieved because all the interfaces/messages between agents were clearly defined and designed. As a result, the procedure of specification, implementation and integration evolved smoothly and efficiently. UBIMATE is now fully integrated and is operating satisfactorily. At the moment, exhaustive tests regarding the functionality and robustness of the platform are being performed, covering all possible circumstances and scenarios, with results so far being absolutely encouraging.

Therefore, the principal aim of designing and implementing an operational prototype and proposing an added value service over 2.5 or 3G mobile networks has been successfully achieved. Following that, commercial exploitation of such a service could be feasible for both hospitals and mobile operators. Because of the Open Architecture used in the development of UBIMATE, the platform’s adoption by the aforementioned actors would entail minimal effort and cost. Moreover, the decline in mobile hardware prices would soon make the service affordable even to individuals with modest incomes.

To conclude, the system is capable of delivering what it promises, which is to promptly offer medical assistance to citizens in need.

Acknowledgment

This work was supported by the Department of Electrical and Computer Engineering, and more specifically by the Telecommunications Laboratory. Finally, we gratefully acknowledge the support and guidance of our Professor, Prof. Michael Theologou during the design and implementation of the UBIMATE platform.

References

[1] Application-Defined Content Languages and Ontologies. Giovanni Caire (TILAB, formerly CSELT).

[2] *Protégé Ontology development tool. http://protege.stanford.edu/*

[3] Natalya F. Noy and Deborah L. McGuinness, Ontology Development 101: A guide to Creating Your First Ontology, Stanford University.

[4] Hofmann-Wellenhof B. Lichtenegger H. & Collins J. (1998). GPS Theory and Practice. Springer-Verlag, Vienna New York, 4th ed.

[5] Java Communication API for Windows CE http://www.teilo.net/home.php

[6] Foundation for Intelligent Physical Agents. http://www.fipa.org/

[7] JADE Programmer’s Manual http://jade.cselt.it/

[8] JADE-LEAP and LEAP project http://leap.crm-paris.com/

[9] Constructing Intelligent Agents Using Java: Professional Developer's Guide, 2nd Edition by Joseph P. Bigus, Jennifer Bigus, Joe Bigus.

[10] J. Sprinkle, C. P. van Buskirk and G. Karsai Vanderbilt, *Modeling Agent Negotiation* University Nashville, TN, 37203.

[11] GeoTools: Open source mapping toolkit http://www.geotools.org

[12] Siv Hilde Houmb, *Security Issues in FIPA Agents*, Norwegian University of Science and Technology (NTNU).

[13] Andreas. L. Symeonidis, Pericles A. Mitkas, Dionisis D. Kechagias, *Mining Patterns and Rule for Improving Agent Intelligence through an Integrated Multi-Agent Platform*.

[14] Federico Bergenti and Agostino Poggi. LEAP: A FIPA Platform for Handheld and Mobile Devices. In John-Jules Meyer and Milind Tambe, editors, Intelligent Agents VIII, *Proceedings of the Eighth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001)*, number 2333 in Lecture Notes in Artificial Intelligence, pages 436 -446. Springer-Verlag: Berlin, Germany, August 2001.