Ubimedic2: An Agent-Based Approach in Territorial Emergency Management

Elton Domnori, Giacomo Cabri, Letizia Leonardi
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Modena e Reggio Emilia, Italy
{*elton.domnori, giacomo.cabri, letizia.leonardi*}@*unimore.it*

***Abstract*—Healthcare emergencies and large scale disasters are two of the most critic scenarios of the healthcare system. Such scenarios require a support that must be distributed, context-aware, reactive and autonomous. This paper presents an architectural approach for coordination based on the multi-agent technology. In the following, we will show how agents behaviour well fits these requirements and how can be efficiently applied to these problems. In this paper we propose a framework, called Ubimedic2, which enables the implementation of Multi-Agent Systems for emergency scenarios management. The framework is FIPA compliant and it is based on the BDI technology.**

***Keywords*-agent; healthcare; Ubimedic2;**

I. Introduction

In the healthcare system, many of the activities require a large-scale coordination over distributed resources. Activities – such as patient scheduling, territorial medicinal distribution and emergencies – require also a continuous coordination among different departments. The Fist Aid service is a critical activity as it requires a very prompt reaction. In territorial emergencies and large-scale disasters, this becomes a real challenge. In such scenarios, a very quick and coordinated response is necessary in order to reduce the serious damage of people health involved in disasters and to save as many lives as possible.

Considering the criticality of these situations, a prompt response, coordination, cooperation and decision making is necessary. The promptness of the intervention is decisive for the succeeding of the rescue operations. Many are the actors involved in these scenarios, not only medical staff but also the fire department and the police department. All these actors must cooperate in order to reach their target. Decision making must be fast and accurate.

The dynamism of the events is another peculiarity of these scenarios. The situation may change rapidly and the same must be the reaction of these actors. The ability to cooperate with each other must not loose performance. The tasks to perform are complex, different and depend on the situations which sometime are unpredictable.

The development of the technology and the widespread of electronic and digital medical devices, makes already possible to save and exchange digital information. Usually, medical devices have their own memory or are connected with computers that collect and elaborate data. The information collected by these devices can be accessed remotely, even in real-time, using ad-hock networks, the Internet or other communication technologies.

A decision making support can be very useful for the human operator. This can save time, avoid human error and increase performance. In large-scale disasters, many operators come into help from other cites around and have no information about the local hospitals distribution for specific pathologies and the maximum number of patient that they can handle, the best way to reach on hospital and many other useful information.

To meet such requirements, we evaluated the use of the agent technology [1] [2] and, based on their peculiar characteristics [3], we propose a suitable solution to these problems.

In the following, therefore, we propose Ubimedic2, a multi-agent architecture able to organize the complex work of rescue operations, taking the appropriate decisions with high performances.

The paper is organized as follows. In Section II, an analysis of the work present in the literature is done. In Section III we propose a case study to better expose the specific requirements of the considered scenario. In Section IV, we introduce the exploited technologies and in Section V the proposed architecture. In Section VI, we describe the implementation characteristics of the Ubimedic2 framework followed by a simulation in Section VII. Finally, conclusions are exposed in Section VIII and future work in Section IX.

II. Related Work

Recently, many researchers in the multi-agent field are focusing on the application of the agent technology to the suport the healthcare system. This interest is growing every day and is covering different aspects. A detailed analysis of the possibility of using agents to deal with the healthcare problems has been done in [4] [5].

The use of agents has been proposed in coordinated hospital patient scheduling [6] [7] [5] [8], in remote monitoring patient [9] [10] [11], in decision support system

PervasiveHealth 2011, May 23-26, Dublin, Republic of Ireland

DOI 10.4108/icst.pervasivehealth.2011.245988

[12] [13] [14], in healthcare knowledge based [15] [16], and advanced health and disaster [17].

In particular, [17] focuses on the triage problem in a disaster scenario that is the same scenario that we are going to consider. It uses electronic devices to collect vital parameters like blood pressure, heart rate and oxygen quantity in blood (using a device called oximeter). The architecture is organized in three levels: *i)* embedded system (i.e., electronic devices aimed to collect vital parameters), *ii)* personal servers (devices aimed to collect the data from the embedded system) and *iii)* central server which is the central operative centre where all the data are collected, stored and elaborated. Some of the weakness of this architecture are: *i)* this system uses only Wi-Fi (which is limited in space) connections to exchange data, therefore two or more devices need to be close to each-other in space because they cannot communicate in large distances and this limits the possibility to operate in real-time; *ii)* the architecture is centralized and there is no interaction between operators; *iii)* there is no support to decision making and no communication with the hospitals. In our work, we are trying to extend this idea and complete it with real-time and large scale communications, self-organization of the components and a decision making support.

A similar work has been done in UbiMedic [18] [19] in order to create a framework that makes possible the communication between devices in a distributed client-server approach. The embedded devices (server side), enabled to collect health information from the patient, are accessible from other devices (client side) such as notebooks and PDAs. This framework provides a controlled access based on a set of policy roles. Ubimedic only focuses on data exchange and collection. Our work starts from this idea, extending it with an autonomous coordination mechanism with the aim to offer better services to patients in a disaster scenario.

Therefore, Ubimedic2 extends the previous Ubimedic system extending the functionalities and empowering with new intelligent components. It has more autonomy and decision making support. The motivations to develop this architecture are deeply exposed in [20].

III. Case Study

To better understand the scenario we are taking into account, let us have a look to what currently happens if there is a car accident involving many people. Someone calls the First Aid Assistance, which has an operative centre, and asks for help. With the guide of a human operator, he/she describes the event and gives as much information as possible about the place and the people involved. Sometimes, the information is vague and not precise but this depends on the person who makes the call. After the operator has collected the necessary information about the event, he/she decides which ambulances to send to the target and calls by phone or by radio one by one the ambulances giving the information collected previously. Moreover, the operative centre must contact the Fire Department and the Police Department to ask for support. Figure 1 illustrates the current communication architecture between the operative centre and the mobile units (such as ambulances, police cars and fire trucks), hospitals and other departments. The mobile units cannot communicate directly with each-other but only through the respective operative centre.

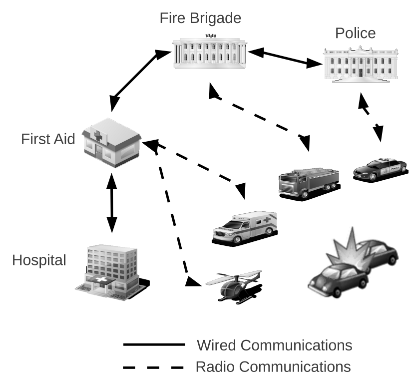

Figure 1. Current communication architecture.

Once the medical staff reaches the place, it can collect more detailed and precise information and communicate to the operative centre if other equipment is needed, such as more ambulances, medical helicopter. During the rescue operation, the medical staff collects the single patient information such as pathology and health parameters (such as heart rate, percentage of oxygen in blood and blood pressure) and writes them in some structured paper sheet for each patient. Currently, this information is not uniform and depends on the qualification of the staff (medical vs. non medical) and the associations that operate in the territory (i.e., Red Cross). The information collected must be exchanged with other staff operators, operative centre and hospital triage.

After the patient receives the first aid, he/she must be sent to the most appropriate hospital. The dispatch is done taking into account the seriousness of the patient pathology, the kind of treatment he/she must receive, the number of patients a particular hospital can support, etc. All these variables make the decision very hard to take in real-time. It must be done in collaboration between the operative centre and the staff present in the place who better knows the condition of the patients. If the available mobile units are not sufficient to face such circumstances, the operative centre must inform a set of standby medical staffs that can be alerted if necessary.

The scenario described above gives an idea of the tasks that must be performed during a rescue operation. The weaknesses of this approach are the way communications occur and the lack of decision making support. The number of communications is large and is done by phone or radio. This means only one communication at a time and this can be a real bottleneck. The leak of support to decision making can bring to errors and takes a lot of time when performed by a human operator. Information is collected in paper sheets making the data transfer difficult and not error free. It gives no way to statistics, digital data storage and information tracking.

Analysing all these aspects during a disaster rescue operation and being aware of the criticality they carry, we propose a new approach able to support a large number of communications, digital data storage and transfer, and decision making support, thanks to the agent-based technology. In the following sections, we will introduce the Ubimedic2 architecture and explain in detail its components.

IV. Exploited Technologies

To make clear the final goal that our system must achieve let us highlight the requirements that must fulfil. Starting from the weak points mentioned above, the new approach must be able to collect and store digital data so to enable a fast and error free information sharing. This information must be shared using electronic devices that can communicate remotely using the appropriate network. This information must be used from intelligent components offering a decision making support to the human staff.

In the scenario we are taking into account, these intelligent components must be aware of the environment they are placed in and must be able to catch any change that may happen. They must be able to react promptly to these changes, and considering their final goal, calculate the most appropriate task to do. These components must be autonomous and able to take decisions. They must be able to communicate with other components for a better organisation but still be self-sufficient if they are in an isolated environment.

To face such requirements, we exploited the software agent technology. An agent is a piece of software, placed in an environment and able to perform autonomous actions in order to reach its target. Agents are *reactive* (perceive the environment changes and act promptly), *proactive* (they take the initiative to undertake an operation) and *social* (they interact with other agents). To better fit our requirements we use the BDI (Believe – Desire – Intention) model [21] to represent agents. For an agent, believes are retrieved from the environment, one or more desires represent its target and the intentions are the set of actions that the agent is going to undertake. Agents can communicate with each-other using FIPA standard messages.

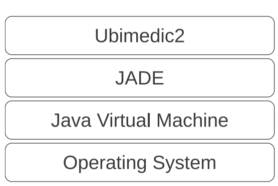

Figure 2. Architecture stack

In Figure 2 the architecture layer stack is shown. Over the Operating System and the Java Virtual Machine we find JADE (Java Agent Development), a free and open-source platform for the development of multi-agent systems. It has been implemented and is distributed by Telecom Italia and is supported by Java 1.4 and newer versions. What makes JADE a good choice is the wide support that is obtaining with smart-phones and in particular with Android OS.

Ubimedic2 is a multi-agent framework, implemented in Java on JADE, whose architecture will be introduced in section V.

V. Architecture

In the scenario we are analysing, there are many actors (human operators) operating in it. First, the human operator at the Operative Centre collects the preliminary information, decides the number of ambulances necessary and which of the available ones must be sent. Then, the ambulances have their medical staff that gives the first assistance to patients, collects information about their pathology and decides the most proper hospital. The Operative Centre must decide the hospital according to the hospital ward related to the patient pathology and the free resources of the hospital so to not saturate it. The health parameters are measured by digital devices (such as ECG and oximeter), analogue devices or from medical observation. All these parameters and observations are necessary to evaluate the patient pathology (or pathologies if there is more than one) and its seriousness.

In our architecture, all the actors and devices that operate autonomously in this scenario are each represented by an agent. These agents will communicate with each-other collecting information, collaborating and taking decision for the operations to undertake. This agent behaviour describes the real life behaviour of the on-site operator where the Beliefs are the information about the patient, the Desire is to bring the patient to the most appropriate hospital and the Intention represents the way the operator will undertake in order to succeed.

In the following, we will introduce the organisation of agents and the way they communicate.

A. Agents

In Ubimedic2, agents represent the control centre operator, medical staff operators, digital devices, mobile units (such as ambulances, medical cars) and stationary units (such as hospital or temporary first aid camp). The agents that represent medical devices are the most simple in our system. They use the client-server approach. When another agent asks for information or service, they answer by fulfilling the request or denying it if the demanding agent is not faithful or has not the requested rights to access that data. The other agents are more complex and autonomous. They substitute human operators in decision making so they must have the necessary information and logic in order to decide the proper operation to undertake. This is possible due to the strict protocols that have been written in order to organize the operations.

Human operators collect the necessary information and operate on the basis of these protocols. In the real life, protocols are necessary to avoid human errors and to rid of responsibilities if things go wrong. Such protocols can be translated into instruction sequences giving to agents even a margin of self decision in case of unpredictable situations. In the following, there is a categorization of all the agents according to their functions.

Service agent (SA) is the agent that represents a device that can be a medical device, notebook or PDA that a human operator uses to insert or read data. These devices have limited functionalities and there is no decision making. Based on their use, the corresponding agents are divided into two categories:

- *Device Service Agent (DSA)* is the agent that represents the medical device in use by the operators. This agent interfaces with the other agents providing a given service or information. It simply responds to a request by returning the requested data.
- *Client Service Agent (CSA)* is the agent that represents the client interface device of an operator such as notebook, PDA and smart-phone. This agent retrieves the information from the client device and collects information from other DSAs to be visualized.

Operative Agent (OA) is the agent that represents the operative unit like ambulance, medical car, hospital, etc. This agent makes use of devices through SA collecting data about the patient. It communicates and collaborates with other Operative Agents in order to take decisions about the patients distribution in different hospitals. We split it into two categories:

- *Mobile Operative Agent (MOA)* is the agent that represents a mobile unit like: ambulance, medical car, helicopter, etc. This agent communicates with service agents (representing devices mounted on the vehicle) in order to receive the necessary information and with other mobile and stationary agents in order to organise the operations.
- *Stationary Operative Agent (SOA)* is the agent that represents a stationary unit such as hospital or temporary first aid camp. This agent communicates only with mobile operative agents. It receives the requests to accommodate the patients with a given pathology and communicates the availability of free resources.

The *Activator* is a special agent in charge of activating the other agents when operative units are ready to be used and of deactivating them when the operative units are no more available. It receives the requests from the operative centre for new operations and dispatches them to the MOA. It asks for the activation of standby units when necessary. It receives requests from ambulances for particular unit activation such as medical helicopter or the intervene of police department or fire department.

In Figure 3 there is a simple representation of the new architecture, agent representation and communication.

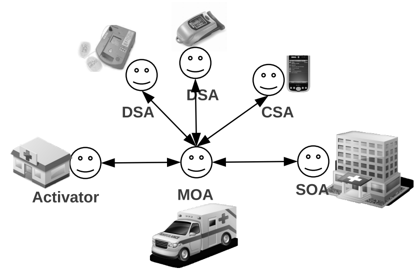

Figure 3. Agent and communication

B. Communication

The agents communication occurs using different technologies as shown in Figure 4. A service agent communicates with one operative agent providing necessary information. The environment where they communicate is limited in space. Devices placed inside an ambulance are used by the ambulance itself. They can use a Wi-Fi or Bluetooth technology to exchange information.

The operative agent uses both GSM and Wi-Fi technology to communicate with other operative agents. The used technology depends on the distance between the operative agents. The ambulances that operate in the same place can communicate faster using the Wi-Fi technology. However, when they need to communicate with units that are far away from the place (i.e., hospitals), the operative agents use the GSM technology.

The RF technology is not taken into account because of the slow data transfer speed and high error rate. It can be used anyway for voice communication between human operators in distance but is not suitable for data.

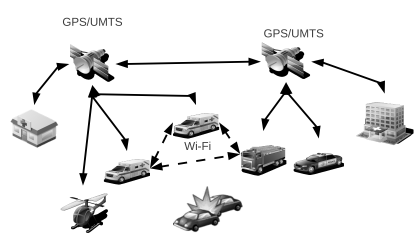

Figure 4. New communication model

VI. Implementation

In the following, we will discuss more in detail about the system implementation including agent behaviour, interfaces, data flow and storage. All the agents implemented in our system must inherit one of the implemented agent classes described above in the architecture, and in particular DSA, CSA and MOA. They extend the basic functionalities of the framework classes to better represent the behaviour of devices and units.

`Ecg` is an agent that inherits from the DSA class and represents a medical device called ECG (Electrocardiogram). This agent is provided with an interface with two input fields, `heart hate` and `fibrillation type` (see Figure 5.A). These two parameters will simulate the data retrieved from the medical device and enable us to perform experiments. When the agent is activated it registers its service to the JADE yellow pages and, when requested, sends the data collected to the operative agent (mobile or stationary). In the same way the Oximeter agent has been implemented (see Figure 5.B).

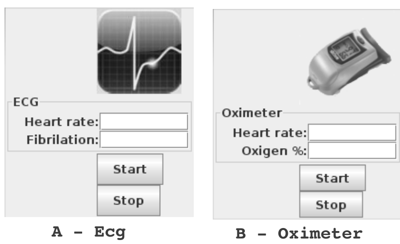

Figure 5. ECG (A) and Oximeter (B) device interface

`Pda` is an agent that inherits from the CSA class and represents a PDA device or a smart-phone. This device is used from medical staff to input information about the patient and read data from medical devices. The interface provides three screen tabs as shown in Figure 6. The first tab is used to input personal data such as patient name, surname and date of birth. The second tab is used to input health information. Data are structured based on the IRC (International Rescue Council) guidelines. The third tab is used to retrieve and display data from different devices.

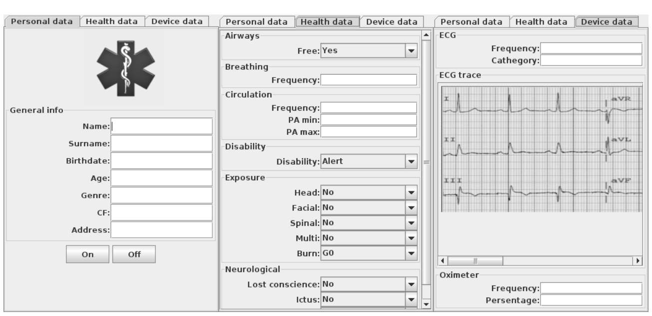

Figure 6. PDA GUI

`Ambulance` is an agent that inherits from the MOA class. Its behaviour is structured based on the finite-state model. It has 5 predefined status (1 – reaching the place, 2 – on-site, 3 – reaching the hospital, 4 – at the hospital and 5 – free). When it receives a new task request from the Activator it accepts only if the status is "5-Free" (sending back to the Activator a positive response), otherwise it is considered as busy (sending back to the Activator a negative response). All data (sent from Activator about a new task and from other devices) are collected from the Ambulance and stored in a database. The `Ambulance` agent is able to decide the main pathology of the patient and the seriousness of it. Moreover, it decides which is the most proper hospital to accommodate the patient. To make such evaluation, health parameters data are retrieved from the database and evaluated. Based on the pathology and its seriousness, the ambulance takes the decision where to bring the patient. If the seriousness is high, the ambulance choose the nearest hospital because time is a very precious element in such cases, otherwise it chooses the most specialized hospital for that pathology.

The `Ambulance` interface is shown in Figure 8 whose details are explained in the next section. It is divided into three sections: A) `State`, which shows the status and the location of the ambulance, B) `Service`, which shows the task information sent from the Activator and C) `Destination`, which shows the hospital where the patient is going to be transported, the main pathology and the seriousness of it.

We highlight that the goal of our project is not to make health diagnosis. This functionality of the Ambulance agent can be extended and improved with the help of medical staff. We did a simple implementation of this method for experimental reasons. We use two functions, `getPathologyAndGravity()` and `getHospital()` for the evaluation that can be re-implemented without any change in our architecture.

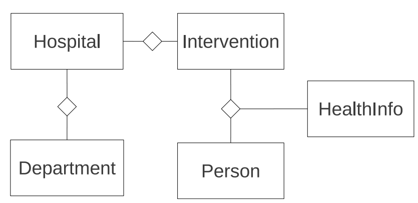

Figure 7. Reduce ER schema

The database is implemented using MySQL. In Figure 7 the E/R schema is shown. The information about the hospitals and the wards it contains, is set up before each agent is activated and can change while it is active. When a new temporary first aid camp is available, the Activator informs the agents about the new hospital, its location and the pathologies that can be treated by defining the wards it contains. Information about hospital distribution can change if ambulances from nearby regions come into help in case of large disasters and need to know the local hospital distribution. The Intervention table contains the data sent from the Activator to the Ambulance agent when a new task in assigned. The `Person` and `HealthInfo` tables are used to store information about the personal details and health information.

VII. The usability of the system

To better understand how the system works, we will introduce a scenario of a first aid intervention as that of Section III and by using a simulation we explain in details the role of agents, their tasks and behaviour. It can be observed how the system reacts after changes has been detected and how agents respond to such changes.

Each agent is provided with a custom graphical user interface in order to access its data and to interact with it. For the simulation, the user is able to set preliminary data (collected from medical devices or medical staff) and change them so to observe and evaluate how different agents behave or change their behaviour in time when they detect a data change from their environment.

In order to simulate GPS location and unit distribution, we will use a two-dimensional coordination space. The distance between the ambulance and the hospital will be calculated as the Cartesian distance between two points. A GPS class will be used to simulate the ambulance path towards the hospital.

Let us suppose that an accident happened and a car crashed into a bike rider. The driver, after offering the first aid, observed that the bike rider suffers of shortness of breath so decide to call the First Aid department. The operator of the First Aid operative centre collects information about the place, the dynamics of the accident and the patient health. He/she decides the pathology of the patient and inserts a new task that will be sent to the Activator.

The Activator, after receiving a new task, assigns it to one of the available ambulances. Considering that all ambulances have the same medical staff, the Activator tries to assign the task to the closest ambulance available. The Activator sends a request to Ambulances, starting from the closest one to the most distant. When an Ambulance accepts the new task, the Activator sends the necessary information to reach the place, the pathology and the seriousness of the patient. If the seriousness is high, the ambulance driver can make use of car siren and car flashing to have priority in case of traffic.

The Ambulance accepts the task only if its status is free that means it is not busy with another task. It receives the information from the Activator (see Figure 8.A) and as soon as it starts, it changes the status from “5-Free” to “1-Reaching the place”. When the ambulance reaches the place, the Ambulance agent changes the status to “2-On site”.

When the medical staff reaches the place, it gives the first aid to the patient. It collects information about the patient and one operator of the medical staff makes use of smart-phone to insert these data through a custom interface, as shown in Figure 6. The first data to be inserted are the personal information such as name, surname and date of birt. These data are sent to the Ambulance agent that stores

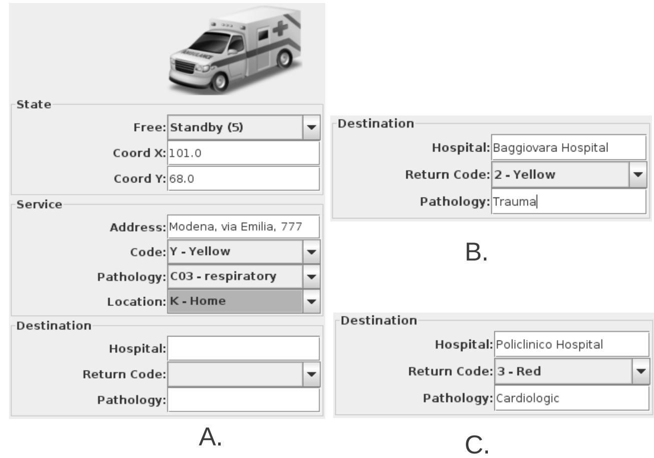

Figure 8. Ambulance agent interface with intervention information (A), first decision (B) and second decision (C)

this information into the database by inserting a new record to the `Person` table. If this information is not available, a fictitious name must be inserted. Other medical observations are inserted into the dedicated interface in Health display tab of Figure 6. This information will be sent to the Ambulance agent, which will store it into the `HealthInfo` table.

Meanwhile, medical staff makes use of medical devices such as ECG and Oximeter to collect other useful information about the patient. When the devices are turned on, the Ecg agent and the Oximeter agent are activated and register their services. Then they are ready to send to the Ambulance agent the information collected. The “Start button” (see Figure 5) in the interface starts the message sending to the Ambulance agent with the collected information. The “Stop button” stops the message sending. The frequency by which these messages are sent is configurable and can be changed according to the requirements. The Ambulance agent stores these data in the same `HealthInfo` table. The medical staff can display these data to the smart-phone using the dedicated section (see Device display tab in Figure 6). In the interface are displayed the most recent data collected by the Ambulance agent.

After collecting all the information about the patient, the Ambulance agent takes the decision where to bring the patient. Several pathologies can be assigned to a patient depending on the health problems. But, only one of them will be considered for the decision of the destination hospital. In this case, pathologies are ranked from the most serious to the less one. The Ambulance, analysing the collected health parameters, checks which pathology prevails.

Let us suppose that all the critical health parameters such as heart rate, blood pressure and oxygen are in range and the respiratory pathology, diagnosed previously by the operator of the first aid department, was due to the high breathing frequency caused probably by the shock of being involved in an accident. The Ambulance agent diagnoses the Trauma pathology and the Green Code for the seriousness.

We placed this scenario in the city of Modena (Italy). There are two hospitals, called Policlinico (close to the centre of the city) and Baggiovara (around 20 km away). Both of the hospitals can treat cardiology and respiratory problems considering the seriousness of these pathologies and only the Baggiovara Hospital hosts a specialized trauma centre.

As all the critical health parameters are in range and time is not a decisive element for the patient treatment, the Ambulance agent decides to assign the most proper hospital to treat the patient pathology. As shown in Figure 8.B, the hospital assigned is Baggiovara, the pathology is Trauma and the seriousness code is Green.

When the medical staff has picked the patient up and is ready to go, the Ambulance agent changes its status to "3-To the hospital". The time the ambulance takes to reach the hospital is simulated by the GPS class (as mentioned previously). Let us suppose that during this time one of the health parameters of the patient changes. Let us use the Ecg agent GUI to change one of the parameters assigning a value out the range. We can change the heart rate to a value over 120 or the fibrillation type to *Atrial de-fibrillation*.

The value detected from Ecg agent is sent to the Ambulance agent which stores it in the database. The next evaluation done by the Ambulance agent (in our experiment the frequency of re-evaluation is set to 20 seconds) will change the main pathology and the seriousness code for the patient. This means a new evaluation for the destination hospital. In this case the Ambulance agent re-evaluates the destination hospital not taking into account the most appropriate one but the nearest hospital. The agent calculates the distance between the ambulance location and the hospitals and decides which of them is closer. Let us suppose that we change the health parameter immediately after the ambulance started. In this case the new choice for Ambulance agent is the Policlinico hospital (see Figure 8.C). If we would wait some seconds while the GPS class would simulate the ambulance close to the Baggiovara hospital, the Ambulance agent would not change the hospital destination as the one chosen previously, in this case it is also the nearest.

In this example, we considered only one ambulance and one patient in order to explain the work-flow of the activities. In case of many interventions or a single intervention with many people involved, each ambulance will follow the same procedure observed previously.

In case of connection problems and isolation of the Ambulance agent from the Hospital agent and the Activator agent, this will not bring to a task fail. The Ambulance agent contains all the necessary information to succeed in its task and is completely autonomous. The lack of communication with other agents may lead to a not proper use of resources. As soon as the communication is re-established, the Ambulance agent can change its behaviour if necessary bringing the patient to another hospital. Agents can communicate traffic jam or other particular situations that may cause problems while reaching the hospital.

VIII. Conclusion

First aid assistance is a complex process that requires high degree of coordination and cooperation among different entities in a short time. The current used approach fails in: communication which is done via phone or radio, no digital information collection is done and there is no way to do statistics analysis and the communication is not error free.

In this paper, we proposed a new approach based on the agent technology to manage territorial emergences. Ubimedic2 is a multi-agent system that supports the human operator in their activities. This system offers a decision making support, real-time data collection and communication. It is full-distributed and scalable. The system is platform independent and is supported by most of the commercial digital devices for communication such as computers, thin clients and smart-phones. Using the digital communication technology reduces time and data preserves their integrity as they cannot be altered during propagation. Decision making made by agents is possible due to the well defined operation roles which the human operators must follow. The operative centre is no more a bottleneck in communication. The time evolution of the events is in the vicinity of minutes so, computation time of distributed agents is quite performing in our scenario.

Simulations has been done to test the usability of the system. They are reliable with respect to a real implementation as the agent behaviour does not change. Device agents will not collect information from test interfaces but from the device itself and the management of the these data remains the same to the one already observed. In the testing we have focussed in data represented using ASCII format such as text and numbers but JADE handle messages with byte content so device agents are able to exchange images, videos or other byte format. An example of application is the ECG trace.

IX. Future Work

Ubimedic2 is a multi-agent framework implemented in Java using the JADE platform. Agents use FIPA compliant messages to exchange data and to coordinate with each other. Further, extension of Ubimedic2 is proceeding into three directions: a) PIM technology integration; b) smart-phone applications, and c) use of ontologies.

PIM (Process Integrated Mechanism) [22] technology is implemented by the Institute for Human and Machine Cognition laboratory (Florida, USA). The idea of this technology is to coordinate a set of robots using a single process that moves through the robots controlling their behaviour. We aim to integrate it with our system and use it to coordinate the operative agents. Using a single process will decrease the number of communications between agents so it should increase performance.

Smart-phones using Android OS support the JADE technology. We are working for the development of applications which will host the Pda agent (already seen in the Implementation section) to test performance and network accessibility.

Moreover, we plan to use ontologies developed especially for health care applications as in [23] in order to increase the interoperability with other systems that use the same ontology.

REFERENCES

[1] M. Wooldridge, *An Introduction to Multiagent Systems*, 2nd ed. Wiley, 2009.

[2] G. Weiss, Ed., *Multiagent systems: a modern approach to distributed artificial intelligence*. Cambridge, MA, USA: MIT Press, 1999.

[3] A. Moreno and C. Garbay, "Software agents in health care," *Artificial Intelligence in Medicine*, vol. 27, no. 3, pp. 229–232, 2003.

[4] J. Nealon and A. Moreno, "Agent-based applications in health care," in *Applications of Software Agent Technology in the Health Care Domain, Whitestein Series in Software Agent Technologies, Birkhuser Verlag*. Verlag, 2003, pp. 3–18.

[5] D. Isern, D. Sánchez, and A. Moreno, "Agents applied in health care: A review," *I. J. Medical Informatics*, vol. 79, no. 3, pp. 145–166, 2010.

[6] K. Decker and J. Li, "Coordinated hospital patient scheduling," in *ICMAS*, Y. Demazeau, Ed. IEEE Computer Society, 1998, pp. 104–111.

[7] M. Hannebauer and S. Müller, "Distributed constraint optimization for medical appointment scheduling," in *Agents*, 2001, pp. 139–140.

[8] C. Heine and S. Kirn, "Adapt at agent.hospital - agent based support of clinical processes," in *ECIS*, 2004.

[9] J. E. Larsson and B. Hayes-Roth, "Guardian: An intelligent autonomous agent for medical monitoring and diagnosis," *IEEE Intelligent Systems*, vol. 13, no. 1, pp. 58–64, 1998.

[10] G. Laleci, A. Dogac, M. Olduz, I. Tasyurt, M. Yuksel, and A. Okcan, "Saphire: A multiagent system for remote healthcare monitoring through computerized clinical guidelines."

[11] M. I. Bagüés, J. Bermúdez, A. Burgos, A. Goñi, A. Illarramendi, J. Rodríguez, and A. Tablado, "An innovative system that runs on a pda for a continuous monitoring of people," in *CBMS*. IEEE Computer Society, 2006, pp. 151–156.

[12] S. S. 0002, B. I. Ismail, F. Haron, and C. H. Yong, "Architecture of agent-based healthcare intelligent assistant on grid environment," in *PDCAT*, ser. LNCS, K.-M. Liew, H. Shen, S. See, W. Cai, P. Fan, and S. Horiguchi, Eds., vol. 3320. Springer, 2004, pp. 58–61.

[13] H. González-Vélez, M. Mier, M. Julià-Sapé, T. N. Arvanitis, J. M. García-Gómez, M. Robles, P. H. Lewis, S. Dasmahapatra, D. Dupplaw, A. Peet, C. Arús, B. Celda, S. V. Huffel, and M. L. i Ariet, "Healthagents: distributed multi-agent brain tumor diagnosis and prognosis," *Appl. Intell.*, vol. 30, no. 3, pp. 191–202, 2009.

[14] D. Isern, D. Sánchez, and A. Moreno, "Hecase2: A multi-agent ontology-driven guideline enactment engine," in *CEEMAS*, ser. Lecture Notes in Computer Science, H.-D. Burkhard, G. Lindemann, R. Verbrugge, and L. Z. Varga, Eds., vol. 4696. Springer, 2007, pp. 322–324.

[15] Z. I. Hashmi, S. S. R. Abidi, and Y.-N. Cheah, "An intelligent agent-based knowledge broker for enterprise-wide healthcare knowledge procurement," in *CBMS*. IEEE Computer Society, 2002, pp. 173–178.

[16] M. Croitoru, B. Hu, S. Dasmahapatra, P. H. Lewis, D. Dupplaw, A. Gibb, M. Julià-Sapé, J. Vicente, C. Sáez, J. M. García-Gómez, R. Roset, F. Estanyol, X. R. Palou, and M. Mier, "Conceptual graphs based information retrieval in healthagents," in *CBMS*. IEEE Computer Society, 2007, pp. 618–623.

[17] T. Gao, T. Massey, L. Selavo, D. Crawford, B. rong Chen, K. Lorincz, V. Shnayder, L. Hauenstein, F. Dabiri, J. Jeng, A. Chanmugam, DavidWhite, M. Sarrafzadeh, and M. Welsh, "The advanced health and disaster aid network: A light-weightwirelessmedical system for triage," in *Transactions on Biomedical Circuits and Systems*, 2007, p. 203.

[18] G. Cabri, F. D. Mola, and R. Quitadamo, "Supporting a territorial emergency scenario with services and agents: A case study comparison," in *WETICE*, 2006, pp. 35–40.

[19] F. D. Mola, G. Cabri, N. Muratori, R. Quitadamo, and F. Zambonelli, "The ubimedic framework to support medical emergencies by ubiquitous computing," *ITSSA*, vol. 1, no. 1, pp. 15–26, 2006.

[20] E. Domnori, G. Cabri, and L. Leonardi, "Coordination issues in an agent-based approach for territorial emergence management," *FINA*, 2011.

[21] A. S. Rao and M. P. Georgeff, "Bdi agents: From theory to practice," in *ICMAS*, V. R. Lesser and L. Gasser, Eds. The MIT Press, 1995, pp. 312–319.

[22] R. Quitadamo, D. Ansaloni, N. Suri, K. M. Ford, J. F. Allen, and G. Cabri, "The pim: an innovative robot coordination model based on java thread migration," in *PPPJ*, ser. ACM International Conference Proceeding Series, L. Veiga, V. Amaral, R. N. Horspool, and G. Cabri, Eds., vol. 347. ACM, 2008, pp. 43–51.

[23] "Snomed: http://www.ihtsdo.org/snomed-ct/."