A multi-agent approach for 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*—The management of territorial emergencies and disasters has become a real issue in the healthcare system. The recent events such as earthquakes and inundations have brought into attention the need for an improvement of rescue operations in order to limit the lost of human lives. Such scenarios require a distributed, context-aware, reactive and autonomous support.**

In this paper we present an architectural approach for coordination based on the multi-agent technology. In the following, we will show how agents behaviour well fits to these requirements and how they can be efficiently applied to these problems.

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

I. INTRODUCTION

Rescue operations during a large-scale disasters have always been a real issue for the healthcare system. In this context, the situations are unpredictable and the number of resources is limits compared to the number of people involved. The First Aid department must supply a prompt reaction and an efficient use of resources in any situation. The rescue process in one of the most critical in the healthcare system. It requires a very quick and coordinated response in order to reduce the serious damage of people health involved in disasters and to save as many lives as possible.

Rescue operations need a prompt response, coordination, cooperation and decision making. The promptness of the intervention is decisive for the succeess of the rescue process. The number of human staff and units to coordinate is likely to be very high. The cooperation and coordination occur not only among medical staff but it must include the fire department and the police department. All these actors must cooperate and coordinate in order to reach their target. They decisions must be fast and accurate.

In these scenarios, the events are characterized by a strong dynamism. The situations can evolve by increasing the number of people involved or covering larger areas. In a more restrict point of view, the health condition of any single patient may change getting more serious. The tasks to perform are complex, different and depend on the situations which sometimes are unpredictable. Despite this, the ability to cooperate and coordinate with each other must not loose performance.

The recent development of the technology offers appropriate tools to support communication and medical analysis. Wired and wireless communication architectures and digital medical devices can give a precious contribution to human rescue process. 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, even in real-time, remotely by using ad-hock networks, the Internet or other communication technologies.

Considering the complexity of the rescue operations and the large number of variables that need to be taken into account, a decision making support can be very useful for the human operator. This will reduce the decision making time and prevent human from errors. Moreover, in large-scale disasters, many of the human operators come into help from other cites around and have no information about the local department distribution for specific pathologies and the maximum number of patient that they can handle, the best way to reach the hospital and many other useful information.

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

Therefore, in this paper we propose Ubimedic2, a multi-agent architecture able to organize the complex work rescue operations, taking the appropriate decisions in a flexible way. 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 Ubimedic2 architecture. In Section V, we describe the implementation characteristics of the framework followed by a simulation in Section VI. Finally, conclusions are exposed in Section VII and future works in Section VIII.

II. BACKGROUND

The multi-agent paradigm has been considered to face healthcare's problems since several years ago. 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 [2] and in [3].

The use of agents has been proposed in coordinated hospital patient scheduling [4] [5] [3] and [6], in remote monitoring patient [7] [8] [9], in decision support system [10] [11] [12], in healthcare knowledge based [13] [14], and advanced health and disaster [15].

In particular, [15] 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 systems (i.e., electronic devices aim to collect vital parameters), *ii)* personal servers (devices aim to collect the data from the embedded systems) and *iii)* the 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 [16] [17] 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) like notebooks, PDAs, etc. that can request to access the data. 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 framework extending it and tries to solve the main healthcare problems that arise in the coordination of a lot of departments and institutions 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 [18].

III. A CASE STUDY

In this section we will give a detailed description of a possible scenario focusing on its relevant aspects. Let us have a look to what happens if there is a car accident. Someone calls the First Aid Assistance asking for help. The human operator collects as much information as possible about the place and the people involved. Information may be 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, 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. Figure 1 illustrates the current communication architecture between the operative centre and the mobile units, hospitals and other departments such as fire and police department.

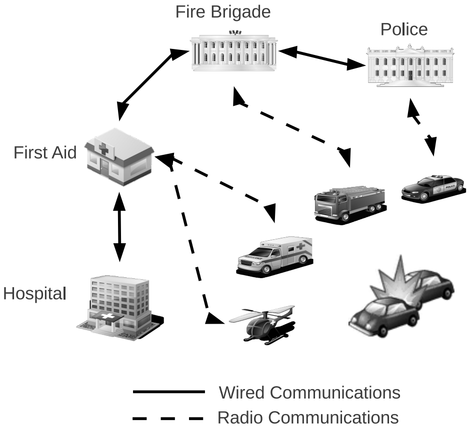


Figure 1. Current communication architecture.

When the medical staff reaches the place, collect more detailed and precise information and communicate to the operative centre if other equipment is needed. During the rescue operation, the medical staff collects the single patient information such as pathology, health parameters (heart rate, percentage of oxygen in blood, blood pressure, etc.) and writes them in some structured paper sheet for each patient. The information collected must be exchanged with other staff operators, operative centre and hospital triage.

The patients receive the first treatment on site and later are sent to the most appropriate hospital for further check and long treatment. The dispatch is done taking into account the patient pathology and its seriousness and the hospital available resources. The decision must be done in collaboration between the operative centre and the staff present in the place who better knows the condition of the patients.

The scenario described above gives an idea of the tasks that must be performed during a rescue operation. The weaknesses of this approach are *i)* the number of communications is high and is done by phone or radio making this process slow and not free from errors; *ii)* no support to decision making is present, and *iii)* information is collected in paper sheets making the data transfer difficult and not error free.

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 communication, digital data storage and transfer, and decision making support thanks to the agent-based technology. In the following sections, we will introduce the exploited technologies used to design the Ubimedic2 architecture.

IV. ARCHITECTURE

In the scenario we are taking into account the new approach must be able to collect and store digital data so to allow a fast and error free data sharing. The data must be used from intelligent components that are able to offer a decision making support to the human staff. 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. Ubimedic2 is a multi-agent framework, implemented in Java using the JADE platform, which architecture will be introduced in the following.

The reference scenario presents many actors (human operators) operating on 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 the department related to the patient pathology and the free resources of the hospital so not to saturate it. The health parameters are measured by digital devices (such as ECG, oximeter), analogue devices or from medical observation. All these parameters and observations are necessary to evaluate the patient pathology (or pathologies if there are more than one) and its seriousness.

In the proposed 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 in the BDI model 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, a categorization of all the agents according to their functions is given.

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, we divided that kind of agent 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, smart-phone. This agent retrieves the information from the client device and collects from other DSAs information 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 each-other in order to make decision about the patient 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 SAs (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 like hospital or temporary first aid camp. This agent communicates only with mobile operative agents when it receives a request to accommodate a patient with a determined 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 2 there is a simple representation of the new architecture with agent representation and communication.

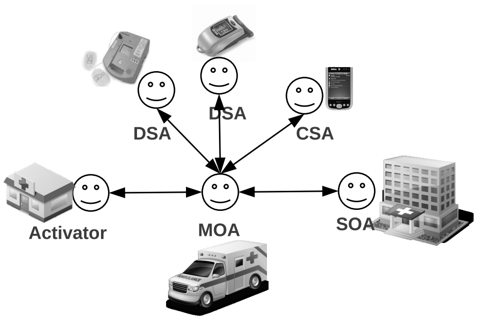


Figure 2. Agents and communication in Ubimedic2

B. Communication

The agents communication occurs using different technologies as shown in Figure 3. A service agent communicates with one operative agent and provides the information/service the operative agent requires. 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 among 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.

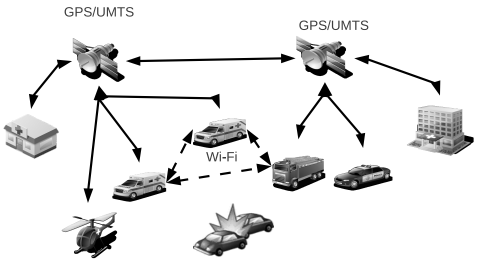
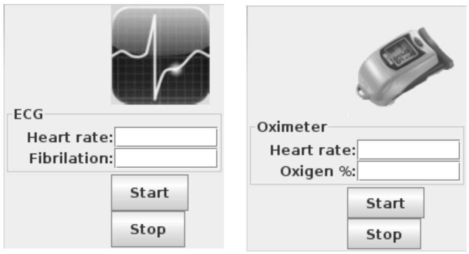


Figure 3. Ubimedic2 communication model

V. IMPLEMENTATION OF THE AGENTS

Here we will discus more in detail about the system implementation including agent behaviour, interfaces and 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 device and units.

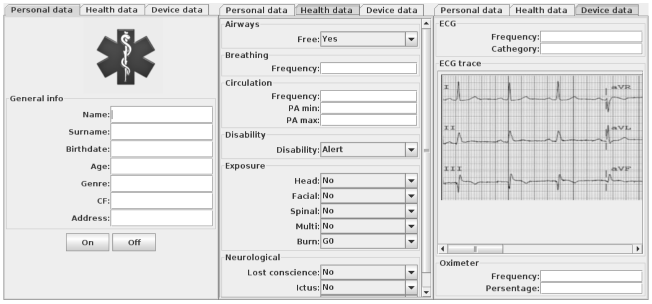
`Ecg` is an agent that inherits from the DSA class and represent a medical devices called ECG (Electrocardiogram). This agent is provided with an interface with two input fields, `heart hate` and `fibrillation type` (see Figure 4.A). These two parameters will simulate the data retrieved from the medical device and allow us to perform experiments. In the same way the Oximeter agent (see Figure 4.B) has been implemented.



A - Ecg **B - Oximeter**

Figure 4. 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 5. The first tab is used to input personal data such patient name, surname, 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.



Free: Yes
Disability: Alert
Head: No
Facial: No
Spinal: No
Multi: No
Burn: G0
Lost conscience: No
Ictus: No

Figure 5. PDA GUI

`Ambulance` is an agent that inherits from the MOA class. Its behaviour is structured based on 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 from the Activator a new task request it accepts it only if the status is “5-Free”, otherwise it is considered as busy. 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 its seriousness, and the most proper hospital to accommodate the patient.

The `Ambulance` interface is shown in Figure 6 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 want to highlight that the goal of our project is not to make health diagnosis. This functionality of the Ambulance agent could be improved with the help of medical staff. We did a simple implementation, for experimental reasons, of the methods `getPathologyAndGravity()` and `getHospital()`. These methods can be re-implemented without any change in our architecture.

VI. TESTING

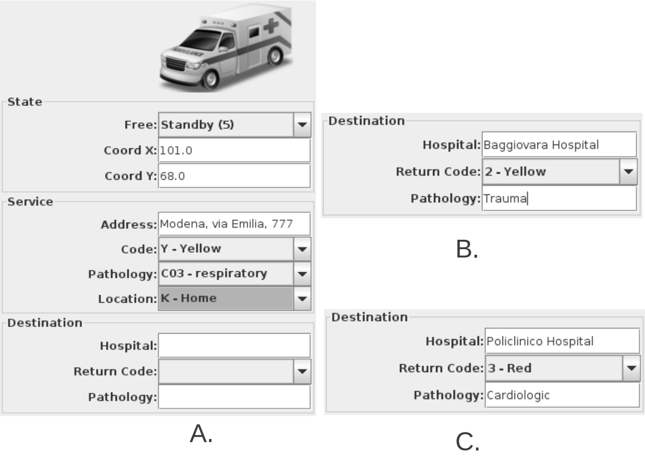
To better understand how the system works, we will introduce a scenario of a first aid intervention as that of Section III and by using a simulation we explain in details the role of agents, their tasks and behaviour. It can be observed how the system reacts after changes have been detected and how agents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

Each agent is provided with a custom graphical user interface in order to access its data and interact with it. In order to simulate GPS location and unit distribution, we will use a two-dimensional coordination space.

Let us suppose that an accident happened and a car crashed into a wheeler. The driver, after offering the first aid, observed that the wheeler 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 health of the patient. He/she decides the pathology of the patient and inserts a new task that will be dispatched from the Activator.

The Activator assigns the new task to one of the available ambulances. The Activator tries so assign the task to the closest ambulance available. The Activator sends a request to Ambulances, starting from the closest 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 6.A).



Free: Standby (5)
Coord X: 101.0
Coord Y: 68.0
Address: Modena, via Emilia, 777
Code: Y - Yellow
Pathology: C03 - respiratory
Location: K - Home
Hospital:
Return Code:
Pathology:

Hospital: Baggiovara Hospital
Return Code: 2 - Yellow
Pathology: Trauma

Hospital: Policlinico Hospital
Return Code: 3 - Red
Pathology: Cardiologic

Figure 6. Ambulance agent interface with intervention information (A), first decision (B) and second decision (C)

When the medical staff reaches the place, it gives the first aid to the patient. It collects information about the patient and inserts these data through a custom interface, as shown in Figure 5. These data are sent to the Ambulance agent that stores this information into the database by inserting a new record to the `Person` table. Other medical observations are inserted into the dedicated interface in Health display tab of Figure 5. 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 are ready to send to the Ambulance agent the information collected. The “start button” (see Figure 4) in the interface starts the message sending to the Ambulance agent with the collected information. The “stop button” stops the message sending. The medical staff can display these data to the smart-phone using the dedicated section (see Device display tab in Figure 5). In the interface, the most recent data collected by the Ambulance agent are displayed .

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 problem. 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) where 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 department for trauma treatment.

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 6.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 to 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 of 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 6.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 connections problems and isolation of the Ambulance agent from the Hospital 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.

VII. CONCLUSION

In this paper we described an approach based on agents to manage territorial emergences. First aid assistance is a complex process that requires high degree of coordination and cooperation among different entities in a short time. This approach offers a solution to the weaknesses of the traditional one.

We proposed a multi-agent framework, called Ubimedic2, which supports the human operator in the rescue activities. This system offers a decision making support and real-time data collection and communication. It is fully-distributed, scalable and platform independent. Data are exchanged autonomously and preserves their integrity. The decision making support is enabled by the well defined protocols. The operative centre is no more a bottleneck in communication. The computation time of distributed agents of the system is compliant with time evolution of the events. Passing from the simulation to a real implementation does not require changes in the agent behaviour, so this makes simulations reliable.

Our approach offers a good solution for the management of large-scale disaster. The system is distributed, autonomous and adaptive, and covers the needs to improve the current organisational approach.

VIII. FUTURE WORK

Our future work focuses in three different directions. The first one is the integration of the PIM [] technology in Ubimedic2 in order to support the coordination among the Operative Agents. This technology has been developed at the Institute for Human and Machine Cognition laboratory (Florida, USA) and the basic idea was to coordinate a set of robots using a common process, called Coordination Process. This process moves from one node to another, using the strong mobility and executes on one node per time effecting the node decisions of the node. In our case, we intend to integrate the dispatching logic inside the process in order to decrease the number of communications between agents.

A second direction is the implementation of interfaces on smartphones using Android OS (supporting the JADE technology) and the test of performance and network accessibility.

Moreover, we plan to use ontologies developed especially for healthcare applications as in [20] in order to increase the interoperability with other systems that use the same ontology.

REFERENCES

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

[2] 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.

[3] 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.

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

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

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

[7] 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.

[8] 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.”

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] “Snomed: http://www.ihtsdo.org/snomed-ct/.”