

# Managing Territorial Emergencies with Ubimedic2

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**—We propose Ubimedic2, an agent based framework that enables the implementation of Multi-Agent Systems for emergency management and territorial disasters. In this demo we present a simple simulation to better expose the Ubimedic2 functionalities.

**Keywords**—agent; healthcare; Ubimedic2;

## I. INTRODUCTION

The First Aid service, territorial emergencies and large-scale disasters are some of the most critical activities in the healthcare system. The criticality of these situations needs a prompt response, coordination, cooperation and decision making. The promptness of the intervention is decisive for the succeeding of rescue operations.

The actors involved in these scenarios, are not only ambulances and medical staffs but also the fire department and the police department. All these actors must cooperate and coordinate in order to reach their target.

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 and coordinate with each other must not loose performance. The tasks to perform are complex, different and depend on the situations, which sometime are unpredictable.

In this demo we will show Ubimedic2 [1] a multi-agent framework able to organize the complex work of rescue operations, taking the appropriate decisions. It is written in Java and relies on the JADE platform. We took the idea of device communication exposed in Ubimedic [2] adding the intelligent component able to self organisation.

## II. ARCHITECTURE

In our architecture, all the actors and the devices that operate autonomously are each represented by an agent. These agents will communicate with each-other collecting information, collaborating and taking decisions for the operations to undertake. In the following, we will introduce the organisation of agents in Ubimedic2.

In Figure 1 there is a simple presentation of the agents and the devices they are installed in.

*Device Service Agent (DSA)* is the agent that represents the medical device. It simply responds to a request by returning the requested data.

*Client Service Agent (CSA)* is the agent that represents the client interface device. This agent retrieves the information from the client device and collects information from other DSAs to be visualized.

*Mobile Operative Agent (MOA)* is the agent that represents a mobile unit such as ambulance, medical car, helicopter. This agent communicates with DSAs and CSAs in order to receive the necessary information and other MOAs 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 MOAs when it receives a request to accommodate a patient.

*Activator* is an agent that receives the requests from the operative centre for new operations and dispatches them to the MOAs.

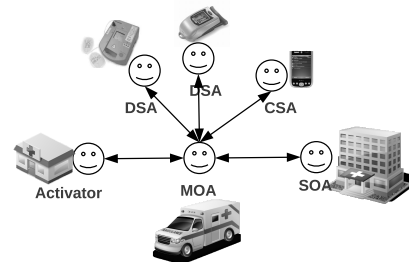


Figure 1. Architecture

## III. TESTING

To better understand how the Ubimedic2 works, we will introduce a scenario of a first aid intervention and, by using a simulation we will explain in details the role of agents, their tasks and behaviour. It can be observed how the system reacts after a change has been detected from the environment.

To perform the test we developed an Ambulance agent (inherited from the MOA class), an Ecg and Oximeter agent (inherited from the DSA class) and a Pda agent (inherited from the CSA class). Each agent is provided with a custom graphical user interface in order to access its data and interact with it. A GPS class will be used to simulate the ambulance and hospital locations.

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 health of the patient. He/she decides the pathology of the patient and inserts a new task that will be sent to the Activator.

The Activator will dispatch it to the closest available ambulance. If the Ambulance agent accepts the new task, the Activator sends the necessary information to reach the place, the pathology and its seriousness.

The Ambulance agent accepts the task only if its status is free. It receives the information from the Activator (see Figure 2.A) and as soon as it starts it changes its status to “1-Reaching the place” and when it reaches the place, changes to “2-On site”.

Figure 2 consists of three screenshots of the Ambulance agent interface, labeled A, B, and C.

**Figure 2.A:** The interface shows a 'State' section with 'Free: Standby (5)', 'Coord X: 101.0', and 'Coord Y: 68.0'. Below is a 'Service' section with 'Address: Modena, via Emilia, 777', 'Code: Y - Yellow', 'Pathology: C03 - respiratory', and 'Location: K - Home'. At the bottom is a 'Destination' section with 'Hospital:', 'Return Code:', and 'Pathology:'.

**Figure 2.B:** The 'Destination' section is updated with 'Hospital: Baggiovara Hospital', 'Return Code: 2 - Yellow', and 'Pathology: Trauma'.

**Figure 2.C:** The 'Destination' section is updated again with 'Hospital: Policlinico Hospital', 'Return Code: 3 - Red', and 'Pathology: Cardiologic'.

Figure 2. 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 health information about the patient and one of the medical staff makes use of smartphone to insert these data through a custom user interface, as shown in Figure 3. Personal information and other medical observations are inserted through the dedicated interface. These data are sent to the Ambulance agent which stores this information into the database.

Figure 3 shows a PDA GUI with three main sections: 'Personal data', 'Health data', and 'Device data'.

**Personal data:** Includes fields for Name, Surname, Birthdate, Age, Genre, CF, and Address, along with 'On' and 'Off' buttons.

**Health data:** Includes sections for 'Airways' (Free: Yes), 'Breathing' (Frequency), 'Circulation' (Frequency, PA min, PA max), 'Disability' (Disability: Alert), 'Exposure' (Head: No, Facial: No, Spinal: No, Multi: No, Burn: G0), and 'Neurological' (Lost conscience: No, Ictus: No).

**Device data:** Includes 'ECG' (Frequency, Category) and 'ECG trace' (a graph showing ECG waveforms). It also includes an 'Oximeter' section with 'Frequency' and 'Percentage' fields.

Figure 3. PDA GUI

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 begin sending to the Ambulance agent the information collected.

The medical staff can display these data to the smartphone using the dedicated section (see “Device data” display tab in Figure 3). In the interface the most recent data collected by the Ambulance agent are displayed.

After collecting all the information about the patient, it takes the decision where to bring the patient. Considering the dominant pathology, the Ambulance agent diagnoses the “Trauma” pathology and the Green Code for the seriousness. As all the critical health parameters are in range and time is not a decisive element, it decides to assign the most proper hospital for the pathology (see Figure 2.B).

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.

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 will change the main pathology and the seriousness code. This means a new evaluation of the destination hospital. In this case the Ambulance agent calculates the distance between the ambulance location and the hospitals and decides which of them is closer (see Figure 2.C).

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

In case of connection problems, the Ambulance agent will still succeed in its task because it contains all the necessary information and is completely autonomous. The lack of communications with other agents may lead to a not proper use of resources. As soon as the communication is re-established, the Ambulance can change its behaviour if necessary bringing the patient to another hospital.

This simulation demonstrates that the Ubimedic2 framework is able to manage the operative units during rescue operations offering support to communication, data storage and decision making. Our approach notably increase the performance during the rescue operations.

## REFERENCES

- [1] E. Domnori, G. Cabri, and L. Leonardi, “Coordination issues in an agent-based approach for territorial emergence management,” *FINA*, 2006.
- [2] 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.