

Supporting Medical Emergencies by MAS ^{*}

Roberto Centeno, Moser Fagundes, Holger Billhardt, and Sascha Ossowski
{roberto.centeno, moser.fagundes, holger.billhardt,
sascha.ossowski}@urjc.es

Centre for Intelligent Information Technologies (CETINIA)
University Rey Juan Carlos
Madrid, Spain

Abstract. In the emerging field of m-Health, advanced applications provide healthcare to people anywhere, anytime using broadband and wireless communications, as well as mobile computing devices. The notion of Service-Oriented Multi-Agent Systems (SOMAS) that has recently been proposed appears to adequately capture the requirements of applications in this field. The THOMAS abstract architecture and software platform supports the construction of SOMAS around an organisational metaphor. In this paper we present an application prototype built on top of THOMAS for a real-world mHealth scenario: mobile medical emergency management in an urban area.

1 Introduction

The use of information and communication technologies to support the efficient provision of healthcare services is an area of huge economic and social potential. In recent years, the notion of m-Health has become prominent, focusing on applications that provide healthcare to people anywhere, anytime using broadband and wireless communications, as well as mobile computing devices [3].

To successfully apply these technologies to this new scenario, a service-oriented approach is gaining popularity [8]. In this context, services are software entities that can be described, published, discovered, orchestrated, and invoked by other software entities. A service-oriented approach to m-health must consider semantics, because in healthcare, every description must have a unique, clear meaning. So, defining and maintaining expressive ontologies for m-health are crucial. Furthermore, healthcare applications are usually based on flexible and complex interactions between people playing different roles in diverse physical and social contexts. Agent technology provides the means to capture this structure because it proposes an interaction-oriented way of designing open software systems [4].

^{*} The present work has been partially funded by the Spanish Ministry of Science and Innovation, grants TIN2006-14630-C03-02 (FPI grants program) and CSD2007-00022 (CONSOLIDER-INGENIO 2010).

Although these research fields have diverse backgrounds and motivations, there is a growing interest in bridging the different worlds: on the one hand, software agents can be viewed as potential users and providers of semantic services, on the other, web service technology can be used to support the interactions in multiagent systems. To this respect, the notion of Service-Oriented Multi-Agent Systems (SOMAS) has recently been put forward [7, 9]. However, very few approaches go beyond a shallow integration, that not only refers to the conceptual level, but also affects the proper mechanisms used in the system [6].

The approach taken by the THOMAS project (MeTHods, Techniques and Tools for Open Multi-Agent Systems) uses the notion of *organisations* as a key abstraction to build SOMAS in open environments. It defines an abstract architecture, as well as a software platform, where agents and services interact according to norms and rules put forward by the organisational structure [12].

In this paper, we describe the application of THOMAS to a case study for a next-generation m-health scenario: the coordination of the different actors (patient, emergency centre, ambulances, hospitals, etc) involved in urban medical emergency management based on wireless mobile technology. Our application is build on top of the THOMAS abstract architecture, and is based on real-world data provided by the Madrid Medical Emergency Service (SUMMA112).

The paper is organised as follows. Section 2 describes our problem domain, the management of medical emergencies in the Autonomous Region of Madrid, and outlines our model of it. Section 3 presents the application prototype designed to deal with these problems. A particular scenario of a medical emergency which has been implemented is presented in section 4. Section 5 concludes our work and points to future lines of work.

2 Semantic Domain Model

Medical emergencies have a high priority given the potential life risk to the patients. These extreme circumstances demand the usage of appropriate resources within an acceptable response time.

In the Autonomous Region of Madrid in Spain, medical emergencies are handled by the SUMMA112. The services provided by SUMMA112 include: reception and management of medical calls, management of beds in hospitals, coordination and assistance of medical emergencies “in situ” and inter-hospital transportation of patients. Human resources consist of approximately 100 call operators, 36 medical doctors and 36 coordination technicians. Material resources include four classes of vehicles for medical assistance: Emergency Mobile Units (Mobile UVI) for advanced life support “in situ”, occupied by one medical doctor, one DUE and two technicians; Rapid Intervention Vehicles (VIR) equipped with technology and medical instruments similar to Mobile UVIs (its purpose is to reach the patient as fast as possible); Home Assistance Units; and helicopters. The SUMMA112 coordination centre receives more than 1.200.000 emergency calls per year – almost one call each 30 seconds. Around 60.000 of these calls are classified as situations of life risk and receive assistance of Mobile UVIs, VIR or

helicopter. The centre’s capability of receiving calls is 3.000.000 per year (8.000 per day).

Out of the various services provided by the SUMMA112, we are particularly interested in the different processes that take place in medical emergency assistance, including the transportation of patients to hospitals. This is one of the most active tasks of the SUMMA112 and certainly one of the tasks with a high social impact. The assistance in medical emergencies is managed by SUMMA112’s coordination centre. The human resources of the centre play the following roles: *Supervisor* is the highest responsible for the assistance operation inside and outside the Coordination Centre of SUMMA112; *Medical Doctor Regulators* regulate the incoming calls, decide which resource to assign given the circumstances, and provide medical aid to the patients; *Technicians* mobilise the resources assigned by the Medical Doctors Regulators; *Operators* are the professionals that receive the calls and catch the first data; *Nursery staff* manage the inter-hospital transportations and bed allocation in the hospitals.

A typical medical emergency assistance starts when a patient calls SUMMA112, asking for assistance. The call is received by an Operator, who gathers the initial data from the patient. Then she forwards the call to a Medical Doctor, who assigns the resources to attend the request according to its evaluation. In some cases the Operator forwards the call directly to the Technician. The Medical Doctor assigns the resource, however it does not mobilise the resource. Technicians do this task by selecting the units taking into account their availability, distance and time to reach the patient, type of resource (ambulances with different capabilities and outsider’s resources, such as *Red Cross*, *Civil Protection*). Finally, according to the patient condition, she is transported to a hospital.

We have modelled the structure of the application domain as a semantic model. The main classes in the model are: *Hospital*, *Patient*, *Vehicle* and *SUMMA*. Other classes, some of them inherited, were modelled in order to specify these main classes. Figure 1 presents a *Protege* screenshot highlighting the Patient class which has the object properties *has-disease* and *has-medical-record*. The remaining properties of a patient are specified through data-type properties.

3 System Architecture

The aim of our system is to provide decision support and value added services to the actors involved in medical emergency assistance procedures, basically, to physicians, patients and coordination staff. There are two factors that have determined the basic decision regarding the system architecture. The first factor consists of the complexity of the underlying application domain and its natural structure – namely a set of autonomous entities that collaborate with each other in order to obtain their goals. The quality of the whole system depends strongly on the efficiency and effectiveness of this collaboration, that is, on the mechanisms used to organise and coordinate the interactions among the active entities. With this in mind, we have chosen an organisation-based multiagent system as the primary architecture. Organisation-based multiagent systems allow for the

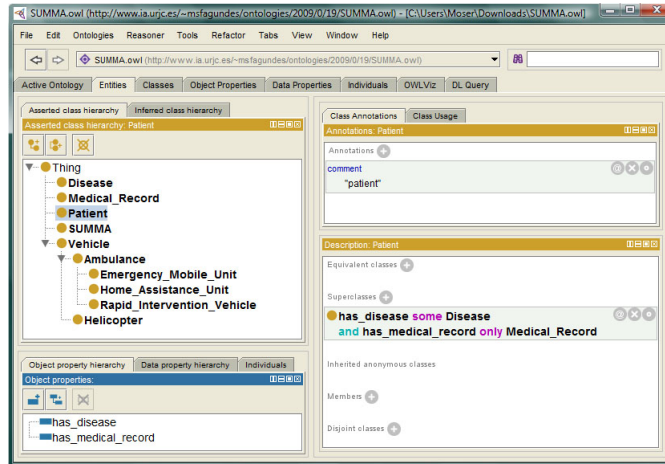


Fig. 1. Classes modelled using the Protege tool

inclusion of organisational mechanisms or elements (e.g., norms, incentive and punishing mechanisms). Such elements assure the functioning of the system and can improve its efficiency by controlling and manipulating the possible interactions that may take place, that is, without considering the actual agents that will participate in the system. The entities that participate in the domain are represented by agents playing predefined roles in an organisation (mHealth organisation).

The second factor we have considered is the fact that identified entities in the application domain can be classified into two classes: *i)* *active entities* whose actions are driven from objectives and that possess a certain degree of freedom in their actions, and *ii)* entities that actually act as providers of certain services (e.g. a medical record store, emergency centre finder, etc.). We propose an architecture where active entities are implemented as agents and non-active entities as web services, and outline its implementation on top of the *THOMAS* platform [1].

3.1 THOMAS Platform

The THOMAS platform implements an abstract architecture for open multiagent systems based on the organisational paradigm. The agents have access to the functionality offered by THOMAS through a range of services included in several modules. The main components of THOMAS are the following:

- *Service Facilitator (SF)*: this component provides simple and complex services to the active agents and organisations. Basically, its functionality is like a yellow page service and a service descriptor in charge of providing a green page service.

- *Organisation Manager Service (OMS)*: it is mainly responsible of the management of the organisations and their entities. It allows to create and to manage organisations.
- *Platform Kernel (PK)*: it maintains basic management services for an agent platform; it is in charge of communication between agents, etc.

Besides the possibility to structure an application through the concept of organisation, THOMAS allows for a seamless integration of multiagent systems with web services. Web services can be easily included and can be registered in the service facilitator.

Figure 2 presents the system architecture using the THOMAS platform. Our mHealth application prototype makes use of the SF, OMS and PK modules of the platform.

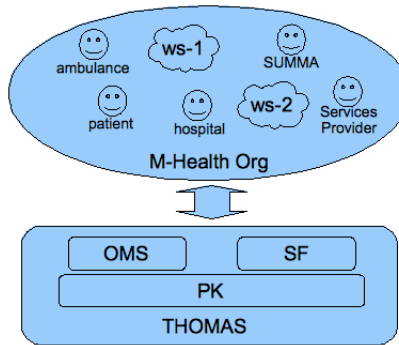


Fig. 2. System architecture

3.2 Domain System Architecture

The whole system has been implemented as an mHealth organisation within the THOMAS platform. We have mapped the active entities of the semantic domain model to roles in that organisation. We defined the roles: patient, hospital, ambulance, emergency coordinator, and “service provider”. The organisation requires exactly one agent to play the role “emergency coordinator”. The other roles can be played by any number of agents. The last role is used to define agents that provide access to web services. The non-active elements of the semantic model have been mapped to the following web services: “medical record store”, “emergency centre finder”, “ambulance finder”, “hospital finder”. All these web services are accessible through “Service Provider Agents” (agents playing the role “Service Provider”).

In our current prototype, the organisation comprises one type of agent for each possible role in the organisation:

- **Patient:** Agents playing role patient represent potential users of medical emergency assistance. Each agent playing the role patient allows the user to contact an emergency centre. For this, the agent uses web services (if available) in order to find the closest emergency centre. If a centre is found, the agent automatically engages an interaction with the agent representing the centre to perform an emergency call. In this call it informs about the current location of the patient and a description of the symptoms (obtained through an user interface).
- **Hospital:** Agents playing this role represent a hospital. Each hospital has different characteristics such as the number of available beds, the set of diseases the hospital is able to treat, location, etc. In the scope of the mHealth organisation, hospital agents decide whether a patient should be admitted, taking into account the number of free beds, the dangerousness of the illness, etc.
- **Ambulance:** agents which are playing this role in the organisation, represent an ambulance vehicle together with the human resources assigned to it. In our current prototype, we have not taken into account the different characteristics of the ambulances: all emergency vehicles are assumed to offer the same services. Ambulance agents can receive missions to take patients and to find the “best” hospital for each patient. When a new mission is received, the ambulance has to pick up the patient, to decide upon the “most adequate” emergency centre for her treatment (usually the closest hospital that provides the medical services needed), and to transport her to this destination. The ambulance agent has to inform the emergency coordinator when a mission is accepted and at the same time it can obtain medical history data of the patient from appropriate web services (if they are available). This functionality may be very useful because it allows the physician travelling in the ambulance to analyse the patient’s medical history before they actually arrive at the patient’s location.
- **SUMMA:** This agent plays the emergency coordinator role which is a main piece in the organisation. An agent playing this role is able to receive emergency calls from patients, and to find the “best” ambulance for each case, thus performing the high-level management of the emergency assistance procedure.
- **Services Provider:** Service provider agents provide access to web services.

Regarding web services, we have implemented simple web service instances for each identified type of service:

- *Emergency Centre Finder:* this service finds the responsible emergency centres for a given location. The result of the invocation of this service will be the identifier of an agent which is playing the emergency coordinator role, in our particular case, the SUMMA agent. This service could be external to the organisation.
- *Medical record store:* this web service allows agents to store and retrieve the patients’ medical history data providing user name and password. This service could be external to the organisation.

- *Ambulance finder*: this service can be considered as internal to an emergency coordinator centre. The service may be implemented with different strategies to decide which is the best ambulance to assign to a given patient. In the current stage, we return the ambulance that is closest to the patient’s position.
- *Hospital finder*: similar to the ambulance finder, this service is considered to belong to the organisation. It is used by an agent playing the role ambulance to obtain the most appropriate destination hospital for a given patient. Currently we assign the closest hospital to the patients location that is able to treat the identified disease.

As mentioned before, within the organisation, all web services are made accessible through service provider agents. The web services have been specified using the standards OWL-S [2] and WSDL [10]. This allows for using standard techniques for service discovery, composition, etc. One advantage of implementing the last two tasks as web services is that we can easily compare different strategies to assign ambulances and hospital to patients.

4 Case Study

In order to evaluate our system we have created a spatial environment simulation tool in which the mHealth organisation can be embedded.

4.1 Spatial Environment Simulator

The environment simulator is an independent module that captures key features of our problem domain. It has two fundamental functions: *i*) it allows to set up agents (and thus, organisations), and *ii*) it recompiles information about the actions that take place in the organisation. The simulation tool is composed of a control layer and a graphical interface.

The control layer receives the actions/interactions and updates the scenario according to these actions. The agents interact with the simulator using the FIPA-REQUEST protocol. The message parameters vary according to the intended action and the role played by the agent. The actions are executed asynchronously – in the exact time step the simulator receives the requests.

Before the simulation starts, the participant agents have to register in the simulator. The registration also follows the FIPA-REQUEST protocol, where the agent specifies its name, role and initial location in the message content. Once the simulation begins, the environment is constantly updated. Basically the simulator controls two parameters: time and location. It represents a spatial world where each agent is positioned at a particular location and at a particular point in time.

According to the role played by an agent, it has a set of available actions. Three roles are represented in the physical environment: patients, hospitals and ambulances. Figure 3 illustrates the set of available actions for each role as well

as the simulator design. Patients have only one available action which consists of informing their position in the map in order to receive assistance. The hospitals are able to accept or refuse patients brought by the ambulances, as well as to release the patients. Finally, the ambulances are able to catch patients that ask for help, release patients in the hospitals, and move around the physical environment by specifying a destination position. Complex routes can be composed by specifying a sequence of positions. Figure 3 illustrates the simulator design and the set of available actions for each role.

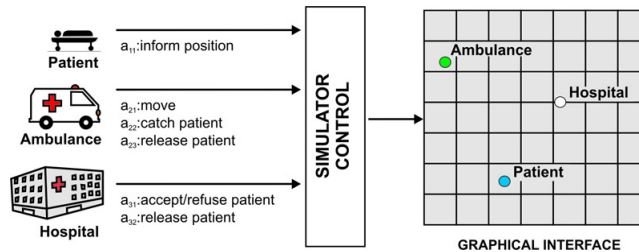


Fig. 3. Simulator design and available agent actions

Figure 4 illustrates the graphical interface by showing an execution where five hospitals (*Escorial*, *Cantoblanco*, *CarlosIII*, *DoceOctubre* and *Fuenlabrada*), one ambulance (*Ambulance1*) and one patient (*Patient1*) joined the mHealth organisation. On the left side of the figure it is shown the *checkbox* for hospital selection. This *checkbox* window is not part of the simulator, but part of an agent launcher.

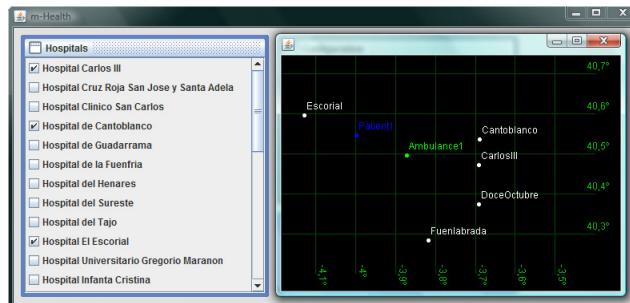


Fig. 4. Simulator screenshot

While the simulator implements real geographical coordinates, at the moment the simulation of travel times is still somewhat simple, as it currently does not take into account the specific topology of the Madrid road network.

4.2 Medical Emergency Assistance Example

We have used the environment simulation tool to carry out several example traces. In order to do so we have set up the mHealth organisation with 31 hospitals and 60 ambulances. The data of the hospitals regarding bed capacities, and ability to treat certain diseases correspond to the hospitals existing in the Autonomous Region of Madrid. In the following, we describe one trace as an example.

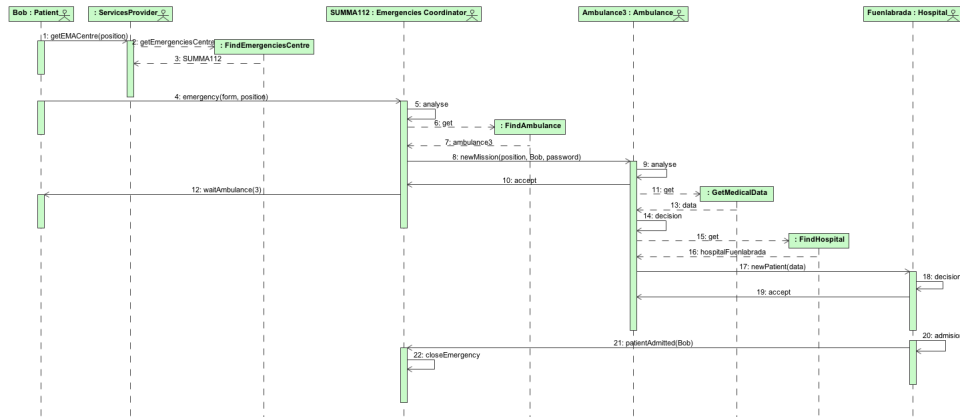


Fig. 5. Case study sequence diagram

Suppose that Bob, a British tourist visiting Madrid, suddenly feels a strong pain in his chest. As this is his first time in Madrid, he does not know what he should do. However, he carries his PDA with the mHealth software suite installed. In this emergency situation, Bob's personal agent joins the *mHealth organisation* playing the *patient* role. Immediately, the patient agent contacts a *services provider agent* looking for an emergency centre finder service.

When this service is found it returns the SUMMA112's identifier which is the agent playing the *emergencies coordinator* role in the mHealth organisation. Together with the SUMMA112's identifier the patient agent receives the form for the patient's data and symptoms. This form is shown to Bob in order to fill it out.

Once Bob selects his symptoms the form is sent to the agent SUMMA112 by the patient agent. When the SUMMA112 receives the new emergency call, it uses its classification method to classify the emergency and to perform a preliminary diagnosis. SUMMA112 detects that Bob is most likely suffering a heart attack. Therefore, he is classified as an urgent patient that needs an ambulance urgently.

The SUMMA112 agent will use its *ambulance finder service* to select the most adequate ambulance to take Bob. Ambulances number 3, 6 and 28 are available closed to Bob's current location. The ambulance finder service will

select *ambulance 3* because it is the closest among all available ambulances in that area.

After an ambulance is selected, the SUMMA112 agent sends a message with the new mission to the *ambulance 3* agent, which is the agent representing *ambulance 3* in the mHealth organisation. When *ambulance 3* receives the new mission it confirms to accept to take Bob immediately.

While the vehicle is going to Bob's position, the *ambulance 3* agent tries to find possible providers of medical data of patients. It finds the service provider agent that implements the *medical record store service* and obtains all available medical history data for Bob. Once the medical history is obtained it is shown to the physician in *ambulance 3*. The physician analyses the data and can determine the required assistance.

When *ambulance 3* arrives to Bob's location, the physicians records his symptoms and perform an "in situ" diagnosis and decide to what hospital he should be taken. In order to do so, the *ambulance 3* agent uses the *hospital finder service* sending Bob's location, symptoms and diagnosis. Taking into account these information and the set of available units in the medical centres, the Fuenlabrada hospital is selected because is the nearest hospital which is able to treat heart attacks.

After that, the *ambulance 3* agent sends a message to the agent representing the *Fuenlabrada hospital* requesting the admission of the new patient. At this moment, the hospital decides to admit Bob because it has free beds and Bob is transferred to the hospital as fast as possible.

Finally, when Bob is admitted at the hospital, the *Fuenlabrada hospital* agent informs SUMMA112 agent about the admission of Bob. At this point, the SUMMA112 agent closes the case successfully.

5 Conclusions

In this paper we have presented an application prototype in the m-health domain. The aim of the application is to provide decision support and value added services to the actors involved in medical emergencies - physicians, patients, hospitals and coordination staff. According to the characteristics of the domain, an organisation-based multiagent system has been chosen where a set of autonomous entities collaborate with each other in order to assist medical emergencies successfully. Along the medical emergency assistance process there exists some entities which just provide certain services without any notion of objectives (e.g. medical record store, emergency centre finder, etc.). In order to best incorporate such services the organisation-based multiagent system has been combined with service oriented computing. With this in mind, our application has been built as an instantiation of the THOMAS abstract architecture – an architecture that combines both technologies, allowing agents to interact with services according to norms, roles, etc., in the scope of an organisation. We have illustrated the dynamics of our prototype by a case study based on real-world data provided by the Madrid Medical Emergency Service (SUMMA112).

Other works, such as [11], have been developed to support medical emergencies processes. In this paper, the authors deal with the problem of selecting the best ambulance to be assigned to an emergency call. They also propose a multiagent approach where each actor involved in a medical emergency event is represented through an agent in an agent society. In addition, the selection of the ambulance is carried out following an auction based mechanism and a trust model based on the past experience with the ambulances' drivers. The main difference with our work is that they focus on the problem of selecting the best ambulance supposing that the hospital is known in advance. Whereas our system intends to provide support to the actors involved in the whole process from its start (the call from a patient) to the end (the admission of the patient in a hospital). Some other work related to disaster management in general has been proposed for instance in [13] dealing with complex scenario where different teams such as polices, fire brigades, ambulances, etc. have to coordinate each other to assist emergencies. In our work we focus on medical emergencies, in a similar way to the work presented in [5], taking into account just hospitals and ambulances.

As future work we pretend to study more complex mechanisms to select the best ambulance and hospital which could optimize the global utility of the system regarding certain desirable parameters (e.g., response time and cost). These mechanisms may take into account aspects like uncovered regions, past experiences with hospitals and ambulances, etc. We would like to compare different mechanisms within different situations (from "normal" functioning to high-load situations such as multiple accidents or other large-scale disasters). In addition, we plan to extend to application in order to provide support also in other typical cases of medical emergency assistance. For instance, in the case a patient is not classified as urgent, the SUMMA112 centre could recommend to visit a hospital soon. The system could provide additional help to that patient, e.g., proposing a route to reach the closest hospital.

References

1. E. Argente, V. Julian, and V. Botti. Mas modelling based on organizations. In *9th Int. Workshop on Agent Oriented Software Engineering (AOSE08)*, pages 1–12, 2008.
2. OWL based Web service ontology. <http://www.daml.org/services/owl-s/1.0>.
3. César Cáceres, Alberto Fernández, and Sascha Ossowski. Cascom - context-aware health-care service coordination in mobile computing environments. *ERCIM News*, (60):77–78, 2005.
4. César Cáceres, Alberto Fernández, Sascha Ossowski, and Matteo Vasirani. Agent-based semantic service discovery for healthcare: An organizational approach. *IEEE Intelligent Systems*, 21(6):11–20, 2006.
5. Anna Ciampolini, Paola Mello, and Sergio Storari. A multi-agent system for medical services synergy and coordination. In J. Nealon, U. Cortes, J. Fox, and A. Moreno, editors, *International ECAI 2004 workshop on Agents applied in health care*, pages 38–46, 2004.

6. Alberto Fernández and Sascha Ossowski. Exploiting organisational information for service coordination in multiagent systems. In Lin Padgham, David C. Parkes, Jörg Müller, and Simon Parsons, editors, *AAMAS (1)*, pages 257–264. IFAAMAS, 2008.
7. Michael N. Huhns. A research agenda for agent-based service-oriented architectures. In Matthias Klusch, Michael Rovatsos, and Terry R. Payne, editors, *CIA*, volume 4149 of *Lecture Notes in Computer Science*, pages 8–22. Springer, 2006.
8. Michael N. Huhns and Munindar P. Singh. Service-oriented computing: Key concepts and principles. *IEEE Internet Computing*, 9(1):75–81, 2005.
9. Michael N. Huhns, Munindar P. Singh, Mark H. Burstein, Keith S. Decker, Edmund H. Durfee, Timothy W. Finin, Les Gasser, Hrishikesh J. Goradia, Nicholas R. Jennings, Kiran Lakkaraju, Hideyuki Nakashima, H. Van Dyke Parunak, Jeffrey S. Rosenschein, Alicia Ruvinsky, Gita Sukthankar, Samarth Swarup, Katia P. Sycara, Milind Tambe, Thomas Wagner, and Rosa Laura Zavala Gutierrez. Research directions for service-oriented multiagent systems. *IEEE Internet Computing*, 9(6):65–70, 2005.
10. Web Services Description Language. <http://www.w3.org/tr/wsdl>.
11. Beatriz López, Bianca Innocenti, and Dídac Busquets. A multiagent system to support ambulance coordination of urgent medical transportation. *IEEE Intelligent Systems*, 23(5):50 – 57, 2008.
12. S. Ossowski, V. Julián, J. Bajo, H. Billhardt, V. Botti, and J.M. Corchado. Open mas for real world applications: An abstract architecture proposal. *Proc. XII Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, 2(1):151–160, 2007.
13. Robocup Rescue. <http://www.robocuprescue>.