Monitoring and Information Fusion for Search and Rescue Operations in Large-scale Disasters

Fabrizio d'Agostino
fdagosti@dis.uniroma1.it

Alessandro Farinelli
farinelli@dis.uniroma1.it

Giorgio Grisetti
grisetti@dis.uniroma1.it

Luca Iocchi
iocchi@dis.uniroma1.it

Daniele Nardi
nardi@dis.uniroma1.it

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 ROMA, Italy

Abstract - *The goal of the project, which is currently under development, is to design tools to monitor the situation after a large-scale disaster, with a particular focus on the task on situation assessment and high-level information fusion, as well as on the issues that arise in coordinating the agent actions based on the acquired information. The development environment is based on the RoboCup-Rescue simulator: a simulation environment used for the RoboCup-Rescue competition, allowing for the design of both agents operating in the scenario and simulators for modeling various aspects of the situation including the graphical interface to monitor the disaster site. Our project is focussed on three aspects: modeling in the simulator a scenario devised from the analysis of a real case study; an extension of the simulator enabling for the experimentation of various communication and information fusion schemes; a framework for developing agents that are capable of constructing a global view of the situation and of distributing specific information to other agents in order to drive their actions.*

Keywords: disaster simulation, multi-agent systems, information acquisition, situation assessment, information fusion, planning.

1 Introduction

Search and rescue of victims in large-scale disasters are not only highly relevant social problems, but pose several challenges from a scientific standpoint. When earthquakes, eruptions or floods happen, a considerable organizational capability to aid the disaster victims as fast as possible is required. However, too often different secondary disasters, connected with the main one, occur, which avoid the correct execution of a rescue plan a priori decided. For example, as reported for the Kobe earthquake (1997) [12], after the earthquake, fires arose between the debris of the destroyed wooden houses, communication infrastructures and transportation systems were largely damaged, causing additional difficulties for the aids.

The goal of the present project is to develop software tools to support the management of this kind of emergency, more specifically to design a support system for search and rescue operations in large-scale disasters, both for prevision and training as well as for operative actions. Even though there are significant results on the design of robots to support search and rescue [14], here we are specifically concerned with the design of a software tool to support both the situation assessment and the planning/control of operations; moreover, we are interested in the deployment of the techniques for achieving cooperation in a multi-agent system [18].

In this context, the RoboCup initiative, the organization which, starting from 1997, arranged the world championship for soccer player robots [9], proposed a new scientific challenge named RoboCup-Rescue [10, 11, 19], where the problems faced are those of bringing aids in a large disaster. The RoboCup objective is to create system based on AI and Robotics technologies, where heterogeneous agents (software, robots, human beings) interact in a cooperative manner. In particular, we are concerned with the possibility of deploying and extending the RoboCup-Rescue Simulator [17], a simulation system whose main feature is the ability to deal simultaneously with many events, thus allowing for the study of different rescue strategies with agents, autonomous or not, and to facilitate the development decision support systems working in real-time.

There are several technical issues that arise in order to pursue the design of a tool like the RoboCup-Rescue simulator: event modeling and simulation, integration and visualization of data, resource management, planning and scheduling, execution monitoring. We are specifically interested in the problems arising in the interaction of a large number of heterogeneous agents [18], i.e. the members of a rescue team. Another relevant research aspect, strictly related to the previous one, is concerned with the coordination of the agents themselves, while they are trying to achieve a common goal [3, 6]. In particular, we address the problem of situation assessment (information fusion) in the disaster scenario, within a multi-agent system.

In order to ground our project in a real scenario we have chosen to access the data of the Umbria and Marche earthquake (1997) and to consider the structures and strategies currently adopted by the Italian VVF (Fire-Dept) [5]. This, not only provides us with a significant body of expertise on rescue operations, but also gives us the opportunity to set up a prototype experimental setting to show the results of the project. The specific elements of the domain under consideration, as well as the relevant features of the RoboCup-Rescue simulator are described in Section 2.

In a complex domain, such as the one of search and rescue operation the agents must show both perception and fusion capabilities as far as the acquisition of information about the operation scenario is concerned; in addition, agents should be able to act reactively as well as to plan and execute complex actions, dealing with failures and continuous changes in the environment. In order to identify the functionalities to be provided to the agents for situation assessment, we have done an extensive survey of the literature on information fusion, specifically looking at the application of multi-agent approaches. The outcomes of this work are summarized in Section 3.

In order to provide the above mentioned capabilities the agent must rely on hybrid achitecture that is both heterogeneous (dealing with different representations of information) and asyncronous (to effectively integrate reactivity and planning) [8]. In Section 4 we describe an agent model and the Agent Development Kit (ADK) that we are developing to support the modeling and implementation of agents embodying the features required for the search and rescue simulation.

We conclude the paper by making some considerations on the exploitation of the proposed approach within Italian VVF and on the research problems that are currently under investigation.

2 The RoboCup-Rescue Simulator: an application to the earthquake of Umbria and Marche

The RoboCup-Rescue Project started in 1999 with the goal of developing a comprehensive urban disaster simulator (see [4]). It aims at producing a software environment useful for both testing intervention strategies in a virtual world and supporting decisions in case of real disasters such as earthquakes or big fires. Below we sketch the overall structure of the simulator to provide some indications on the components that need to be developed in order to apply the simulator to a specific disaster scenario. For a detailed description of the simulator see [17].

The RoboCup-Rescue Simulator has a distributed architecture, formed by several modules, each of them being a separate process running in a workstation on a network. The following are the main components of the simulator:

- *Geographic Information System* - The GIS module holds the state of the simulated world. Before simulation begins, it is initialized by the user in order to reflect the state of the simulated area at a given time, then it is automatically updated at each simulation cycle by the kernel module.

- *Kernel* - This module is connected to any other module. At each step it collects the action requests of the agents and the output of the simulators, merging them in a consistent way. Then, the kernel updates the static objects in the GIS and sends the world update to all the connected modules.

- *Simulators* - Fire-simulator, Collapse-simulator, Traffic-simulator, etc. are modules connected to the Kernel, each one simulating a particular disaster feature (fire, collapses, traffic, etc.). At the beginning of every simulation cycle, they receive from the kernel the state of the world, then they send back to the kernel the pool of GIS objects modified by the simulated feature (for example, a pool of burned or collapsed buildings, obstructed roads, etc.)

- *Agents* - Agent modules are connected to the kernel and represent "intelligent" entities in the real world, such as civilians, police agents, fire agents, etc. They can do some basic actions, such as extinguishing a fire, freeing obstructions from roads, talking with other agents, etc. Agents can also represent non-human entities: for example they can simulate a police-office, a fire station, an ambulance-center, etc.

- *Viewers* - Their task is to get the state of the world, communicating with the Kernel module, and graphically displaying it, allowing the user to easily follow the simulation progress.

In order to use the RoboCup-Rescue simulator in the context of the present project several issues must be taken into account. The first issue we have addressed is the choice of the domain that should be based both on the availability of data and on suitability of the RoboCup-Rescue simulators in modeling such an area. It is beyond the scope of the present project to design specific disaster simulators. After the domain has been identified the representation of the information concerning the disaster scenario to be used in the simulation must be constructed. Then, the features of the agents need to be analyzed to verify whether they are suitable for the modeling of the scenario. While the domain and its representation are described in the rest of this section, subsequentely we consider architectures and agent systems for information fusion, before addressing the design of agents in the search and rescue scenario.

2.1 The earthquake of Umbria and Marche

Throughout the fall of 1997, a serious earthquake affected the italian regions of Marche and Umbria: many housing estates as well as important artistic and religious monuments were heavily damaged, first and foremost the world-famous Basilica of S. Francesco in Assisi. In order to experiment and verify techniques and methodologies developed in our work, we have selected Foligno, one of the most important cities in that region, as an interesting scenario for running a disaster simulation; below we discuss the features of the chosen site and describe the main aspects of its representation within the simulator. In order to build such a representation we have developed a graphical editor that, starting from a bitmap of the site, supports the input of the description.

2.2 Domain features

Foligno is located in a flat region of eastern Umbria. Its urban structure is characterized by a medieval center surrounded by more recent suburbs; in particular we are fixing our attention on an area of about 1 km^2 in the city center.

In the area under consideration there are no high-rise buildings; most recent structures are mid-rise, in the four- to nine-story range. All large, multi-story buildings were constructed of reinforced concrete. Oldest buildings were mainly constructed of rubble-work, whereas only few structures are steel frame buildings or wood buildings. There are no industrial structures; most buildings are housing estates having variously-shaped plants. The road network is quite irregular, with not very large roads and narrow alleys. The following paragraph describes the model we adopted to represent this domain in the simulated world. The following paragraph describes the model we adopted to represent this domain in the simulated world.

2.3 World model

The world model we adopted is derived from the RoboCup-Rescue simulator model; it is somewhat minimal, but it could be easily extended to fit real scenarios more closely. It deals with three main entities or *object classes*: *buildings*, *roads* and *nodes*, respectively. The road network is described by a graph having one or more edges for each road and one node for each crossroad and for each junction between adjacent edges constituting a road. Also, a node can represent a linkage point (*access-point*) between a building and a road. Each object class (*building*, *road*, *node*) is characterized by a number of attributes describing a specific instance of the class. The following paragraphs show the features of each class.

2.3.1 Buildings

Building objects represent every kind of building on the map: houses, police offices, hospitals, fire stations, ambulance centers, refugees, etc. As a result of a earthquake shock, a building can collapse and obstruct a road; moreover, a building can catch fire more or less likely, according to its constituent material; for example, a concrete building is less flammable than a wooden one. Further, buildings can have one or more floors and one or more linkage points with the surrounding roads. The main attributes of buildings are: *Plant*, *Kind*, *Material*, *Fieryness*, *Brokenness*, *Floors*, *Entrances*.

2.3.2 Roads

Road objects are the edges of the road network graph; they represent every street, lane, tunnel, bridge, etc. in the map. A road can be partially or totally obstructed by rubble in consequence of the collapsing of an adjacent building. Further, a road has one or more traffic lanes on each side and can have a sidewalk or not. The most relevant road attributes: *Kind*, *Length*, *Width*, *Block*, *Repair-Cost*, *Lines-to-head*/*Lines-to-tail*, *Sidewalk-width.*

2.3.3 Nodes

Nodes represent crossroads or linkage points between buildings and roads. Moreover, a whole road can be split into two or more adjacent edges, connected to other nodes. The following are the most important attributes of this class: *Roads, Signal, Signal-timing*.

3 Architectures and Agents for Information Fusion

Information fusion is broadly used in various application fields such as defense, geoscience, robotics, health, industry and many techniques have been developed for the fusion process (see for example [7, 20]). Moreover, the degree of abstraction of information (raw data, features, or symbols) that is used in the fusion process is an important design element to take into consideration in the development of an information fusion system. In this work we are mainly interested in the definition of the system architecture that is required for implementing a system of robotic agents acting in a rescue domain like the one described in the previous section.

The design and development of system architectures is a central issue in the design of a fusion system. All the architectures proposed in the literature may be grouped in three categories:

- Centralized
- Hierarchical
- Distributed

The centralized approach to the development of fusion architectures is the most popular in the literature due to the fact that centralized fusion can be characterized as a well defined problem. Data collected from all the sensors are processed in a single central unit that performs the fusion task. This approach is optimal when there are no communication problems (bandwidth, noise) and the central unit has enough computational resources to perform fusion among data. Most fusion algorithms have been developed for the centralized fusion architecture, and many applications have been realized using this approach.

However, in recent years distributed and hierarchical approaches are becoming more popular, thanks to the the spreading of communication technology. Hierarchical fusion architectures are based on different layers of nodes: at the lowest layer, fusion nodes collect data from sensors to perform a first fusion process on these data, then they send their results to a higher layer of fusion nodes. Each of the higher layer nodes collects the results of fusion from lower layers and perform a different fusion process among them. The overall architecture can be seen like a tree where each node is a fusion node and the leaves of the tree are the sensors. For example, in [15] a hierarchical architectures is presented for the design of a monitoring system for a power plant. The architecture is made of two layers of fusion, a first fusion is performed among a subset of sensors, then fusion results are send to the central monitoring system that fuses the results of the sensor subsets. This design allows to send to the central monitor more reliable and smaller data, resulting in an increase in the performance of the whole monitoring system.

Finally, distributed architectures differ from hierarchical ones in the topology of the fusion nodes. Also in this case each node performs locally a fusion process and send the results to other fusion nodes. However, in distributed architectures there are no hierarchical layers, but every fusion node can communicate with each other. The connections are thus arbitrary and the overall architecture can be represented as a graph of fusion nodes. A distributed approach to information fusion is very frequent in agent base systems, see for example [13, 16], that are presented in the next section.

As compared with the centralized ones, distributed and hierarchical architectures have the following advantages:

- Lighter processing load at each fusion node, due to the distribution over multiple nodes.
- Lower communication load, due to the reduction amount of data to be communicated.
- Faster user access to fusion results, due to reduced communication delay.

On the other hand, using distributed or hierarchical architecture requires the development of fusion algorithms that are specialized for those architectures.

In the rescue context, centralized approaches are not suitable, since the hypotheses of perfectly reliable and low-cost communication among the nodes (that in our case are agents acting in the rescue scenario) is not verified. In fact, during rescue operations often communication among the agents is very difficult, noisy, and with low bandwidth and thus a centralized approaches would easily lead to a complete stall of the operations of the agents.

On the other hand, using distributed or hierarchical approaches leads towards the use of multi-agent based approaches to perform the fusion and these approaches are seen best suited for the rescue domain.

Before defining our proposal for the design and development of cognitive agents for rescue operations, we have studied many multi-agent systems for information fusion that can be found in the literature. We report in the next section some of them, highlighting the features that may be of interest in our application domain.

3.1 Multi-Agent approaches to Information Fusion

In the last ten years, the multi-agent technology has became a very important research topic and many researchers in the information fusion field started exploring the possibilities to develop information fusion systems by making use of multi-agent systems. In the following we analyze different works which exploit the agent technology for the information fusion problem.

An important part of the multi-agent system techniques applied to the information fusion problem is the gathering and selection of information obtained as a consequence of a query, for instance to an Internet search engine. Even if the process of fusion is more related to the symbolic level, and thus it is not strictly a sensor fusion problem, this kind of works cover a large portion of the efforts of the information fusion research. The issues which arise while executing queries is to gather relevant heterogeneous results in a coherent manner avoiding their overlapping, and to deal with information sources which are not homogeneous in sharing a common ontology. The work of Zhang and Zhang [21] is one example of this kind of study: the authors present an agent based information fusion system which is used to collect results of the same query from different resources, and to address the decision fusion problem which arise when many agents make different decisions starting from the same information depending on their own knowledge base. The interaction between decisional agents and information retrieval agents described in this work is an interesting issue that we need to integrate in an architecture in the rescue domain, since also in this scenario there are agents that collect information about the status of the world, namely people in the disaster area, and agents that make decisions based on information available.

Aside this class of works, the multi-agent techniques has been applied to multi sensor system to fuse both raw sensor data and feature data. In [13] Knoll and Meinkoehn proposed an agent system architecture to realize a distributed sensor network. They started by analyzing the benefits which a well suited multi-agent system could bring to the development of a distributed sensor network by reducing the amount of data transferred among the sensors and a central fusion unit, and letting the sensor network be easy to expand. The solution they propose is a fully distributed sensor-agent network where every element has the capability to exchange data only with those network nodes which can really contribute to increase the global knowledge by providing useful data. This kind of distributed sensor networks may be useful in rescue domains when it is possible to exploit information of sensors in the environments (e.g. temperature sensors for detecting fires, video cameras for acquiring information about collapsed building or road traffic, etc.) and it is required to integrate this data with higher level information coming from people.

The work of Anderson [1] is focused on a smart use of a distributed sensor network by making use of a multi-agent system. It analyzes the advantages that agents can introduce in the Baltic Watch project framework. The goal of this project is to develop a monitoring structure which can increase the security condition within the Baltic Sea. Also in this work there is an important element to be considered for rescue operations that is the choice of the best communication topology among the agents. In fact, in a rescue scenario, due to the general difficulty in communicating with other agents, it is very important to define communication infrastructures and to ensure that each agent can communicate only with those agents that are able to effectively process and interpret the message.

Finally, the work of Regis et al. [16] is based on the same view of agent which extends a sensor. Their research investigates how a distributed real-time control system can be realized by making use of agents which control distributed sensors and execute the fusion process. They developed agents which have the capability to schedule the different tasks they needed to achieve their goal under soft real-time constraints. The importance of this work in rescue application is the introduction of the autonomy of each agent. In fact, every agent acting in a rescue domain must act in collaboration, but autonomously with respect to the others, in other words even if communications are not available for a period of time the agent must continue his work.

We have specifically considered all these works in order to define the system architecture for our fusion system in the rescue domain that is presented in the next section.

4 Agent design

In this section we propose an agent architecture that is suitable for the modeling of agents with the features required in a search and rescue scenario. Moreover, we define the components of a framework that allows for an easy and structured implementation of agents operating in real-time.

4.1 Agent model

The literature on agent approaches to information fusion shows that agents can be seen both as system components that allows for a modular software approach to system design and modeling concepts. We follow the latter approach thus identifying our system agents with the entities that act in the scenario.

The overall structure of the agent should satisfy the needs for information acquisition as well as the support for intelligent action. In Figure 1 we describe the functional model of the agent.

We sketch the main function accomplished by each process (represented as an oval in Figure 1).

World and World Management - The world is the structure that represents the whole agent knowledge about the world. In order to grant consistency it can only be modified by the *World Manager*, that implements different update policies. For instance, in the rescue domain the world is composed by the rescue world objects such as roads, agents, buildings, etc., plus some information about the agents state.

At each instant the world contains the result of the fusion of the information received since the starting time, and, hence, it contains the knowledge for decision making.

Plan Executor (PE) - A plan may be seen as a graph that specifies the actions to reach a goal. Each node is a state, edges specifies the state transitions caused by action

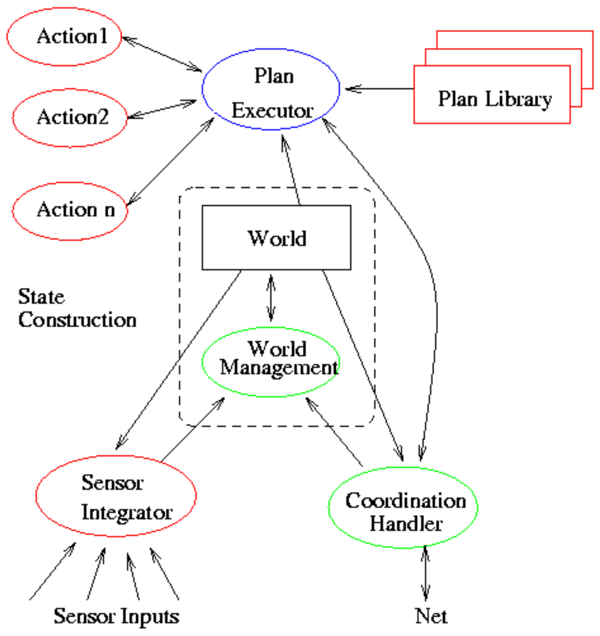

Figure 1 : Functional Agent Description

execution. The Plan Executor performs two tasks:

- It receives from the coordination handler the goal to reach and peeks from the plan library the plan to achieve it.
- It executes a plan by starting or stopping primitive actions at each state, finding which edge to follow in case of a condition branch.

A plan switch can occur after the recognition of a plan failure and/or a goal switch may be forced by the coordination module.

Sensing Integrator - In general, an agent can be equipped with a large variety of sensors and/or sensing capabilities, each one providing different type of input. The task of the sensor integrator is to calculate the new state of the world starting from the previous states and the current sensor info. At present, in the RoboCup-Rescue simulator, agents have limited sensing capabilities, modeling vision and hearing, and not requiring the application of sophisticated integration techniques.

Coordination Handler - In the multi-agent system the communication capability plays a fundamental role. Basically two kinds of information have to be handled:

- information about the sensed world
- information about the action coordination

The information exchanged by the agents about their perception of the world serve as input for the information fusion process. As already noted information can be represented by features or more generally and by symbolic representations. The information concerning the actions coordination is used to decide the *agent* specific goal, allowing the agent system to accomplish the overall goal of agent team.

As argued in the previous section information fusion can be accomplished at several degrees of abstraction and therefore the agent architecture must be heterogeneous in that it should handle different representations. In particular, we aim at combining fusion of raw sensor data (although in the simulator perception is currently not considered at this degree) as well as features and symbols. As shown in Figure 1, through the communication channel agents can exchange information about the world as perceived, and therefore regarded as additional sources of information that must be considered in order to update the representation of the world. In addition, the types of agents range from the operation center to the individual agent, possibly organized in several independent structures. This makes it necessary to provide centralized fusion capabilities as well as hierarchical and fully distributed ones.

In this respect, the proposals that have been presented in the previous section are generally focussed on one specific type of representation, as on a specific architecture, and therefore are not directly applicable in our scenario. To provide the required flexibility, the heterogeneous and asynchronous architecture proposed in [8] for the design of a single agent is adopted to implement a multi-agent system with the capabilities of a distributed organization for fusion and of the integration of information at different degrees of abstraction.

In fact, in this architecure heterogeneity allows for taking into account different level of representation of information acquired by the agents and asynchronicity allows for integrating planning capabilities and fusion processes in a dynamic environment.

4.2 ADK for Robocup Rescue

The Robocup Rescue environment provides a tool (the Rescue ADK [2]) that simplifies the agent construction. However, at present, the above sketched agent model cannot be directly defined on top of the ADK. Therefore, we are designing an extension of the ADK that is suitable for our purposes, by implementing the modules discussed above within our asyncronous architecture.

Moreover, we are concerned with the communication capabilities of the agents. In particular, we are interested in experimenting with communication infra-structures that cannot be directly modeled in the setting supported by the ADK. For example, one cannot specify that a set of agents is linked through a broadcast communication channel.

As specified by the ADK, agents have their own representation of the state (memory), but it does not allow to maintain multiple or uncertain representations of data, which are possibly needed to handle the fusion of data coming through the communication with other agents, and are required by our heterogeneous architecure.

Finally, we are implementing a mechanism for visualizing not only the global scenario, but also each agent view of the world in order to evaluate the process of situation assessment within the multi-agent system.

5 Conclusions

We have presented the current development of the ongoing project "Monitoring and Information Fusion for Search and Rescue Operations in Large-scale Disasters". The aim of the research is to develop a tool to support search and rescue operations in large scale disasters. In particular, we are deploying the RoboCup Rescue Simulator, which has been developed to provide an environment for experimentation of multi-agent technology in the framework of the RoboCup initiative. We have addressed a specific application domain: the disaster scenario recorded after the earthquake of Umbria and Marche. Moreover, we have focussed on the problem of situation assessment and discussed the features of some approaches to information fusion based on multi-agent approaches, that are related to the present project. Finally, we have sketched the agent model that we are developing in order to design the various types of agents.

The availability of the RoboCup-Rescue simulator has been extremely valuable for the development of the present project, providing an experimental setting that can be effectively used for developing a prototype implementation. The simulation can serve both to evaluate various strategies for information acquisition and situation assessment, as well as to make it more understandable to the VVF personnel the potential benefits of an integrated approach to the simulation and monitoring of a real search and rescue scenario. While it is premature to consider the effectiveness of the tool in the management of operation, both the analysis of past scenarios as well as the training of personnel seem to be already suitable for application.

Acknowledgements

This research was supported under contract number F61775-01-WE-030 of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, United States Air Force Research Laboratory.

We also gratefully acknowledge the technical support of the RoboCup Rescue Project.

References

[1] L. Andersson. Intelligent agents. *Master Thesis in informatics*, 1999.

[2] Micheal Bowling. Robocup rescue: Agent developement kit version 0.4, available at [17].

[3] P.R. Cohen, H.J. Levesque, and I. Smith. On team formation. *Contemporary Action Theory*, 1999.

[4] Robocup Rescue Technical Committe. *Robocup Rescue Simulator Manual v0.4*, available at [17].

[5] Pianificazione e monitoraggio in tempo reale dei soccorsi in gravi disastri (in Italian). http://www.dis.uniroma1.it/~rescue.

[6] B.J. Grosz. Collaborative systems. *AI Magazine*, 1996.

[7] D.L. Hall and J. Llinas (Eds.). *Handbook of Multisensor Data Fusion.* CRC Press, 2001.

[8] Luca Iocchi. *Design and Development of Cognitive Robots.* PhD thesis, Univ. "La Sapienza", Roma, Italy, 1999.

[9] H. Kitano and et al. Robocup, a challenge AI problem. *AI Magazine, Spring*, 1997.

[10] H. Kitano and et al. Search and rescue in large scale disasters as a domain for autonomous agents resarch. *IEEE International Conference on System, man and Cybernetics (SMC99)*, Tokyo, 1999.

[11] H. Kitano and et al. Wearable system for the disaster mitigation problem: Mission critical man-machine interface for the robocup rescue simulator. *Proceedings of International Conference on Artificial Reality and Telexistence(ICAT99)*, 1999.

[12] H. Kitano and S. Tadokoro. Robocup rescue, a grand challenge for multiagent and intelligent systems. *AI Magazine*, 2001.

[13] A. Knoll and J. Meinkoehn. Data fusion using large multi-agent networks: an analysis of network structure and performance. *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, (MFI '94)*, pages 113 - 120, 1994.

[14] Murphy, J.G. Blitch, and J.L. Casper. Robocup urban search and rescue events: Reality and competition. *AI Magazine*, In press.

[15] Du Qingdong, Xu Lingyu, and Zhao Hai. D-s evidence theory applied to fault diagnosis of generator based on embedded sensors. *Proceedings of the Third International Conference on Information Fusion (FUSION 2000), Volume 1*, 2000.

[16] B. Horling, V. Lesser, V. Regis, W. Thomas. The soft real-time agent control architecture Technical Report 02-14, University of Massachussets, 2002.

[17] Robocup Rescue Web Site. http://robomec.cs.kobe-u.ac.jp/robocup-rescue.

[18] K. Sycara. Multiagent systems. *AI Magazine*, 19(2):79-92, 1998.

[19] S. Tadokoro and et al. The robocup rescue project: a multiagent approach to the disaster mitigation problem. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA00), San Francisco*, 2000.

[20] L. Valet, G. Mauris, and P. Bolon. A statistical overview of recent literature in information fusion. *IEEE Aerospace and Electronics Systems Magazine*, 16(3):7-14, 2001.

[21] Zili Zhang and Chengqi Zhang. Result fusion in multi-agent systems based on owa operator. *Proceedings of the 23rd Australasian Computer Science Conference (ACSC 2000)*, pages 234 - 240, 2000.