FMSIND: A Framework of Multi-Agent Systems Interaction during Natural Disaster

Muhammad Aslam[1], Muhammad Tariq Pervez[2], Syed Shah Muhammad[2], Seemal Mushtaq[1], Ana Maria Martinez Enriquez.[3]

[1]Department of CS & E, U. E. T. Lahore, 54890, Pakistan,
[2]Department of CS, Virtual University, Shadman Campus, Lahore, 54600, Pakistan,
[3]Department of CS, CINVESTAV-IPN. D.F. Mexico.
tariq_cp@hotmail.com

Abstract: Multi-agent systems have a potential to collaborate with each other using their language but the challenge is to make them work intelligently during the situation of catastrophic disaster. In such situations, it is extremely viable to diagnose and dispose resources like ambulances, volunteers, etc. timely, in order to help out people and reduce casualties. We studied the existing frameworks and methodologies in this area but none of them satisfy the requirements on the whole. If one lacks the coordination between agents then other has deficiency of decision support system. This was a motivation for us to propose a framework that covers all aspects of the problem. In this paper, we propose an algorithm to find out the plans of other collaborative agents for coordination and a complete architecture of the framework. The decision support system has been incorporated in the framework for taking optimized decisions. We take a scenario as a case study to verify and validate the proposed framework. . We also show the implementation of interaction among the agents. [Journal of American Science 2010; 6(5):217-224]. (ISSN: 1545-1003).

Keywords: Agents; multi-agent systems, JADE; decision support system

1. Introduction

The gigantic earthquake of Pakistan (Admin, 2006) disturbed the country not only economically but also with the loss of precious lives, urged us to contribute our skills in order to mitigate the consequences. We could reduce casualties by having the correct information in-time or in advance.

Multi-agent systems (MAS) are useful from the real world applications to the graphical applications e.g., computer games. They are also used for coordinated defense systems. MAS can be very useful in the scenario of natural disaster where multiple physical or logical entities need to coordinate in very critical situation. MAS make the agents to work collaboratively and intelligently, which is the key requirement for the selected application (Sadik et al., 2006).

Traditional rescue teams and disaster management departments take decision, for rescuing people and handling casualties, according to the situations but they are either far-away to sense the situation properly or the geographic factors failed their decision ability. The consequences of this lead them with higher number of casualties and infrastructure loss. By using the MAS systems and the other accessories like sensor system, decision support system, neural system, etc., we can have an infrastructure upfront that have better coordination among the agents. The coordination part can be done using the existing frame-works of agent communication e.g., Java Agent DEvelopment Framework (JADE) (Caire, 2007).

MAS, the Decision Support System (DSS), repositories, and the interaction between MAS are the desired components of Disaster Management System. Scalable Agent Grooming Environment (SAGE) (Sadik et al., 2006) (Farooq et al., 2005) offers scalability and fault tolerance but it lacks DSS and the collaboration between agents. The EQ-Rescue (Fiedrich, 2006) does not have DSS. The Earthquake Management System and Knowledge Oriented Sensors Web (Sia et al., 2008) do not have the collaboration between agents. None of the system is providing the complete solution upfront. We integrate all necessary components in our solution.

In Section 2, we analyze existing systems/frameworks. We present materials and methods for the proposed Framework of Multi-agent Systems Interaction during Natural Disaster (FMSIND) in Section 3. In Section 4, results and discussions are presented. Finally in Section 5, we conclude our work and present the future work.

2. Related Work

In this section, we give a brief review of previously proposed prominent frameworks:

2.1 Scalable Fault Tolerant Agent Grooming Environment (SAGE)

SAGE is a fault tolerant, Scalable and decentralized environment (Sadik et al., 2006) (Farooq et al., 2005). It consists of following components.

Agent Management System (AMS) is the core component of SAGE and has the supervisor level control over the platform and its components. It can either request agents to perform some function or forcefully deploy them on any task. AMS also performs management functions i.e., create, suspend, kill, and resume along with control of various agents' platform parameters.

Directory Facilitator (DF) provides the services of yellow pages. Agents can register their services to DF. They can query DF as well to find what services are provided by the other agents. Agents can also de-register or update their registration contents.

Visual Management Agent (VMA) provides the services of visualization of the platform. It provides the Graphical User Interface (GUI) for administration and monitoring services. Apart from GUI, it provides tools as well that can be used to interact with other components like AMS, DF and to test application agents administratively.

Message Transport Service (MTS) is responsible for sending and receiving messages between system and application agents. The agents involved can belong to single local platform or multiple platforms. There are two modes of communication in SAGE: Inter Platform and Intra Platform.

Agent Communication Language (ACL) provides the message format and the description of the agent intention. It is responsible for the creation of messages that are understandable by all of the components involved in MAS.

SAGE gives the advantage of decentralization and distribution with message priority queue to achieve fault tolerance and scalability. However, in SAGE, agents lack coordination which results in the non -cooperative system that cannot work in timely manner.

2.2 Earth Quake Rescue

Earth Quake Rescue (EQ-Rescue) is a distributed simulation system that models the initial response activities after earthquake disaster (Fiedrich, 2006). It consists of following three models:

Disaster World Model is used for the representation and simulation of disaster area. The simulators in the model have complete knowledge of the actual situation. The information includes the real damage of buildings, infrastructure, roads, the number of injured and trapped persons and the buildings that are on initial stages of catching fire. Therefore, the information world resulting from these simulators is called as complete information world.

Resource Model - Resources in the field have limited access to the information world provided by the Disaster World Model. Each resource has specified radiant in which it can sense the disaster world for pre-defined resolution. The assignment of task to the resources can only be devised by emergency operation center members or agents. The further processing then takes place on the devised assignment by the resource simulator itself.

Emergency Operation Center (EOC) agents or members do not have the complete information about the disaster world. Rather they have access to the Incomplete Information World (IIW) that populates the information from the pre-disaster database. This is the decision making component of EQ-Rescue.

The present version of EQ-Rescue does not include an agent-interface to the decision makers / EOC agents. The communication mechanism between the EOC agents and the working agents is lacking.

2.3 Knowledge Oriented Sensor Web

Knowledge oriented sensor web is also an approach in this stream where agents work intelligently not only to sense the situation but also to take action (Sia et al., 2008). They deal with the updated information as well as the decision making part as per their sense. They use multi-agent system in two parts: Sensing and Response. They link these parts through knowledge-base and the decision support system. The core part of the knowledge oriented sensor web is the interaction between sensor agent (SA) and disaster response agent (DRA). The interaction between these agents depends upon specific environment and is carried out through communication protocol.

But, if the interaction is through web, the communication channel such as gateway is required. The architecture uses rule-based expert system, MAS and Sensor Web; however, it does not describe about the gateway for communication between different components of the system.

2.4 Earthquake Management System

In Earthquake Management System (EMS), policy based migration of mobile agents is used for movement of agents in disaster management system (Sadik et al., 2006). Policies consist of rule-based statements which are the set of pre- post conditions and action statements. These operations have the mobility and task execution strategy. The operations can be picked from the set of operations list, for instance, relocating the agent. The user of the mobile agent picks a goal to perform. The goal first needs to satisfy all condition operations associated with the pre-conditions. Then, control will be handed over to Do-Action part. That will have the mobility strategy and the execution methods. After that post-conditions must need to be satisfied before completion.

The architecture deals only with the mobility and interaction of agents. The other necessary parts like decision support system are lacking in the system. The agents in this architecture lack coordination and collaboration.

2.5 The ALADDIN Project

Autonomous Learning Agents for Decentralized Data and Information Networks (ALADDIN) is a multi-disciplinary project that deals with the dynamic nature of uncertain distributed and decentralized Intelligent Agent Systems (Padhy et al., 2006). The project deals with the communication and interaction between the multiple agents to achieve the individual and collective goal. It concerns with three main tasks:

- How the interaction between agents can be structured?
- How the class of methods and agents can be used to coordinate for solving the problem during the operations?
- How interactions between these agents can be modeled and simulated?

To achieve the above tasks the project exhibits the following methods:

- Auction Method. Resource allocation in MAS is also an extensive research area especially if we come in the domain of natural disasters. Multiple approaches have been used to fulfill this task like the Bidding Strategy, in which there would be an auction over the resource allocation. The project uses neural networks to optimize the resource allocation and to train the system (Enrico et al., 2008)
- Coalition Detection Method. Coordination and interaction between MAS is done through the agent communication languages but in case of multiple MAS, the coordination would be a bigger and complex problem because every MAS would have its own protocol to communicate. Less number of resources and their deployment according to the situation is an enigma. Anytime Coalition Formation is an algorithm which forms the team of resource agents and assigns them weight-age and assists the system to pick the suitable coalition (Rahwan et al., 2007).

3. Materials and Methods

3.1 Framework of Multi-agent System Interaction during Natural Disaster (FMSIND)

The key components of our framework are the following (see Figure 1).

- Sensor System senses the situation (e.g., seismic reading of earthquake or the smoke sensors etc.) and if it is crossing the threshold then it intimates to the Sensor Agents. The sensor devices are used for this purpose e.g., for earthquake detection; the Seismic Reading device can be used, and for fire detection; the usual Fire Alarms can be used.
- Sensor Agents sense the situation on-site. These agents have link with sensor devices that give them activation signals, if threshold limit crosses. The sensor device is responsible for activation of sensor agent. Once the sensor reading crosses the threshold, at the same time, it produces the alarm along with the signals to the sensor system.

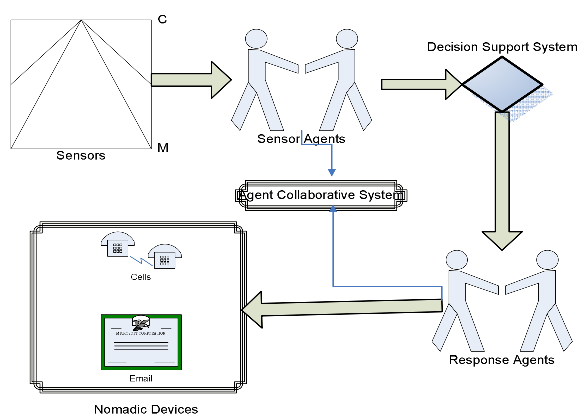

Figure-1. Architecture of FMSIND

- Interactive MAS consists of agent communication framework like Java Agent Development Framework (JADE)
- Decision support system includes learning agents, neural networks, and the components of data warehousing to find the pattern.
- Response agents are activated as per instruction of the decision support system.

Nomadic Devices like personal digital assistants, email systems, mobile phones are used for intimations and warnings for taking actions.

Our main focus in this framework is on the interaction between the agents at disaster site and resource site (like hospitals, fire-brigade offices, rescue services, etc.). The messages between the agents are sent using agent communication language (ACL) (Fornara and Colombetti, 2002). Research in the project of ALADDIN has developed such an algorithm, and it offers several orders-of-magnitude improvement on current state of the art (Rahwan et al., 2007). We use this algorithm in the coordination part of on-site and resource-agents.

3.2 FMSIND Architecture: A Detailed View

This section gives a detailed view of the architecture of FMSIND. The on-site part and the remote-site part are included in the Figure 2. Following are the key components of this architecture:

Sensor Agent (SA) has the responsibility of activating the suspended agents as well as getting information from the sensor system about the loss, and providing this information to the database and decision support system. It also has the responsibility to get data from the decision support system about the on-site required resources.

Supervisor agent has responsibility to get the required resources data from the sensor agent and assign tasks to the on-site coordinating agents. For example, if there is a need of 5 ambulances, the supervisor agent passes the information that the site requires 5 ambulances and assigns duty to the on-site agent to coordinate with the remote-site agents to get this done.

On-site agents get the resources required on on-site from supervisor agents and check their availability by interacting with the disaster database management system. Then, they provide this data to the decision support system to get the plan. They also provide the information about the resources required on on-site to the remote-site agents.

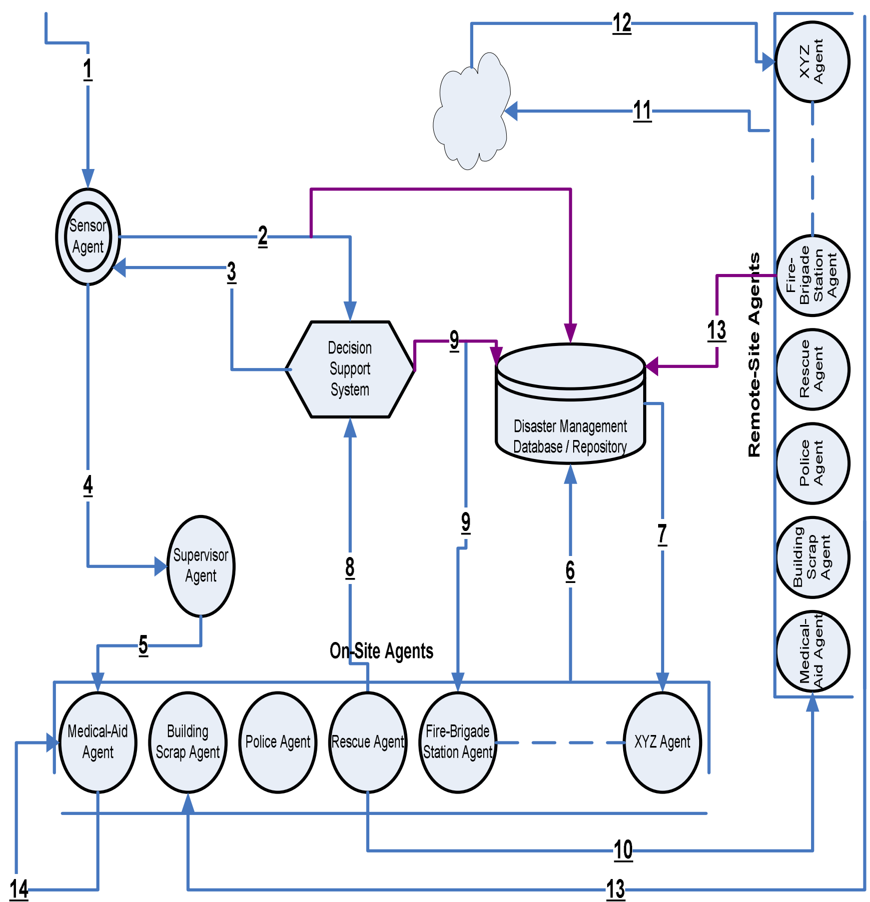

Figure-2. Detailed architecture of FMSIND

Remote-site agents send the required resource information to the resource handlers as per the addresses provided by the on-site agents. The resource handlers send the reply to the remote-site agents. The remote-site agents then compile the information and send back to the on-site agents.

Decision Support System (DSS) has the responsibility to decide about the number of resources required by the site as per the information provided by the sensor agent. We use the decision support system for devising a plan to assign resources to the on-site agents.

Disaster Management Database System / Repository (DB) has the data about disasters, allocated and deployed resources. It also has the data about the resources availability and their addresses. The resource coordinators have rights to update this information in the database.

3.3 Step wise Description of FMSIND

1. Sensor input data: Number of casualties, injuries, building damage area, fire losses, damage of infra-structure, etc.

2. SA sends data to DSS and DB

3. DSS sends number of resources required e.g. ambulances, doctors, beds, volunteers, fire-brigades, policemen etc. to SA

4. SA sends requirement of resources data to supervisor agent.

5. Supervisor agent devises the duties to the relevant on-site agent.

6. The on-site agents check the availability of resources and their addresses

7. DB sends the required data with resources' addresses and their status.

8. The on-site agents send data to DSS to get the resource allocation plan.

9. Resources with allocation plan are sent to the on-site agents and saved in the DB as well

10. A request is made by the on-site agents to the relevant remote-site agents to perform a task.

11. The remote-site agents send a message to the resources' coordinators (RC) on their preferred information media e.g., email, short messages on cell phones etc. to get the required resources.

12. RC allocates the required resources and informs the remote-site agents.

13. The remote sire agents give information about the allocated resources to the on-site agents. For example, how many ambulances and doctors have been sent to the on-site? The remote-site agents also update DB with the allocated resources to the on-site agents.

14. A loop from step no. 6 to onward until the fulfillment of requirements.

3.4 A Detailed Functional Description of the On-Site and Remote-Site Agents of the FMSIND

On-Site Part

The on-site part of our FMSIND comprises of multiple agents, the decision support system and the disaster management repository system. The agents include the sensor, supervisor and categorically named coordinating agents. System starts when sensors cross some threshold and activate / awake the sleeping / suspended sensor agent that has responsibility to signal the agents of the whole system to activate them. Sensor agents get information from the sensor system about the casualties, damage, loss of infrastructure, etc. It compiles the information and sends to the database and to the decision support system as well. The decision support system provides the information about the resource requirement according to the raw-information provided by the sensor agent. The data is then sent to the sensor agent.

The sensor agent provides the resource requirement data to the supervisor agent. The supervisor agent has the responsibility to devise the duties to the relevant category on-site agent. The on-site agents then communicate with the database that has information about the resources availability and their addresses.

This information is sent to the agent back as the results of query and the agent passes the requirement of resources and the available resources information to the decision support system. The decision support system then takes decision how these resources should be allocated and sends the plan to the agent as well as to the database to keep it up-to-date. The on-site agent then coordinates with the remote agent to implement the provided plan.

Remote-Site Part

Agents of this site become operational when any call is sent to these agents by the on-site coordinating agents. The on-site agents advise different duties to these agents according to their operation. For example, on-site rescue agent will send the information to the concerning remote rescue agents to send the number of rescuers on the disaster place.

The remote-site agent has responsibility to directly communicate with the resources' coordinators (RC). They send information to the RC that this number of resources required at the required place. The information is sent to the RC on their preferred information media e.g., the hospital coordinator choose the media Short Message Service (SMS) to coordinate with the agents. The remote-site agent waits for the response from the coordinator. They send intimation again if they get no response after certain time.

After receiving response from RC the remote-site agents update the database with the information that how many resources are available and how many has started work on the devised task. The remote-site agents also update the on-site agents about the information which becomes the cause of starting the operation of on-site agents.

4. Results and Discussions

4.1 Agents Coordination

The coordination part is very important in our architecture of FMSIND. The agents plan to perform a task. This is their default behavior. While working with MAS, they need to collaborate with one another in order to achieve the broad goal of the system. There are two things that come across while going deep into interaction/collaboration:

1. Planning comprises of how to perform a task locally. Each agent by default does planning whenever it needs to perform a task. In MAS they need to know the plans of others in order to perform the tasks in synchronous way.

2. Scheduling is the actual strategy of performing a task with synchronicity. The scheduling is done after devising a plan. The agents need to do the planning on two levels:

- **a.** Local – they need to form their own plan to perform the task
- **b.** Global – they need to know the plans of other collaborative agents for coordination

For the global scheduling and planning we present an algorithm in Figure 3.

4.2 Validation

As a case study we are taking the earthquake of 2005 which came in northern areas of Pakistan and its magnitude was recorded as high as 7.6 (Admin, 2006). The northern areas have a lot of mountains, hills, glaciers, etc. which made the help difficult to reach there. The outside help takes significant delays because of the geographical condition and the hazards on the way to these areas. The rate of casualties increased enormously because of this problem. We place our system and see how we can reduce the casualties and other hazards. Sensor agent, supervisor agent and all other on-site agents such as medical-aid agent, building scrap agent, police agent, rescue agent and fire brigade station agent are installed on the on-site i.e. Muzaffarabad, Azad Kashmir to perform relative duties. Relative remote agents are installed in Islamabad and Rawalpindi.

Suppose the earthquake has occurred. The seismic (Carrington et al., 2008) instrument has touched the threshold value. Sensor agent receives data from the sensor devices, installed on the on-site, and provides it to the decision support system. DSS calculates that the on-site requires 2 fire brigade vehicles. Now the supervisor agent assigns this duty to the fire brigade station agent of the on-site. The on-site fire brigade station agent sends this requirement to corresponding remote fire brigade station agent (as shown in Figure 4). The remote fire brigade station agent sends an email or makes a phone call for a concerned person about this emergency. The concerned person sends the required fire brigade vehicles to the on-site and inform to the remote fire brigade agent as well. The remote fire brigade station agent then informs the concerned agent of the on-site. During this process, database is updated accordingly. This process is automatic, very fast that and the on-site is provided with the required resources immediately and number of causalities and other losses of property is minimized and saved at an enormously rate.

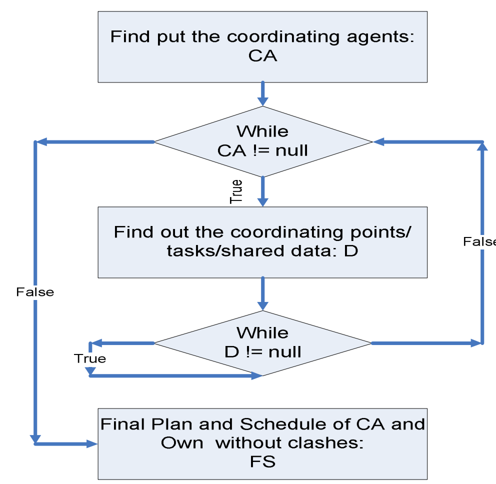

Figure-3. FMSIND Interaction Algorithm Flow Chart

5. Conclusion and Future Work

We integrate all necessary components of a disaster management system into the agent based architecture. The architecture has complete information about what should be included in a disaster management system. We implemented our system's agent interaction part using JADE. The solution is compatible with any system as it is based on Java.

The framework needs to be checked with more real world case studies to test the potential problems of communications and coordination among the agents.

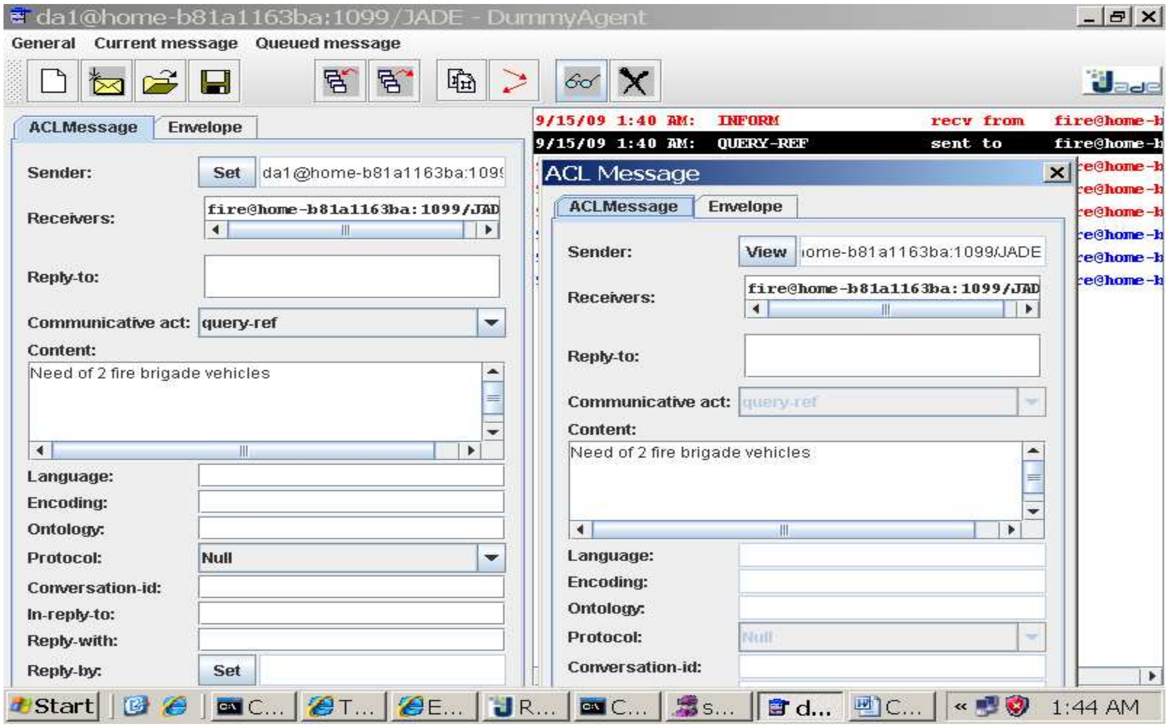

Figure-4. Interaction between on-site and remote site fire brigade agents

Acknowledgements:

Authors are grateful to Mr. Sarfraz Ahmad who reviewed our research work and helped us in learning JADE.

Corresponding Author:
Muhammad Tariq Pervez
Department of CS, Virtual University,
Shadman Campus, Lahore, Pakistan
E-mail: tariq_cp@hotmail.com

References:

1. Admin. Pakistan observes 1st earthquake anniversary. Available at: http://www.pakquake2005.com/. 2005. Index.php? Topic=General, (2006).
2. Caire G. JADE Tutorial. Available at http://jade.tilab.com/, 2007.
3. Carrington L, Komatitsch D, Laurenzano M, Tikir M M, Michéa D , Goff N L, Tromp J. High-frequency simulations of global seismic wave propagation using SPECFEM3D_GLOBE on 62K processors. In Proc. of the 2008 ACM/IEEE conference on Supercomputing, 2008.
4. Enrico H, Rajdeep G, Dash K, Byde A. Optimal bidding strategies for simultaneous Vickrey auctions with perfect substitutes. In Proc. Of 8th Intl. Workshop on Game Theoretic and Decision Theoretic Agents, 2008; 10-17.
5. Farooq H A, Suguri H, Ali A, Malik S, Mughal M M, Shafiq O, Tariq A, Basharat A. Scalable fault tolerant Agent Grooming Environment (SAGE). In the Fourth International Joint Conference on Autonomous Agents and Multi agent Systems (AAMAS), 2005.
6. Fiedrich F. An HLA-based multiagent system for optimized resource allocation after strong earthquakes. In Proc. of the 38th conference on Winter simulation, 2006.
7. Fornara N, Colombetti M. Operational specification of a commitment-based agent communication language. In Proc. of the first international joint conference on autonomous agents and multiagent systems, 2002.
8. Padhy P, Rajdeep K, Martinez D K, Jennings N R. A utility-based sensing and communication model for a glacial sensor network. In Proc. Of 5th Int. Conf. on Autonomous Agents and Multi-Agent Systems, Hakodate, Japan 2006; 1353-1360.
9. Rahwan T, Ramchurn S D, Dang V D, Giovannucci A, Jennings N R. Anytime Optimal Coalition Structure Generation. In 22nd Conference

on Artificial Intelligence (AAAI), Vancouver, Canada 2007.
10. Sadik S, Pasha M, Ali A, Ahmad H F, Suguri H. Policy Based Migration of Mobile Agents in Disaster Management Systems. IEEE - ICET '06. International Conference on Emerging Technologies 2006.
11. Sai X, Li J, Wang Z. Knowledge-Oriented Sensor Web for Disaster Management: From Sensing to Decision Making. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Part B4. 2008.

3/26/2010