Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

AN HLA-BASED MULTIAGENT SYSTEM FOR OPTIMIZED RESOURCE ALLOCATION AFTER STRONG EARTHQUAKES

Frank Fiedrich

Institute for Crisis, Disaster, and Risk Management
The George Washington University
1776 G Street NW, Suite 101
Washington, DC 20052, U.S.A.

ABSTRACT

In this paper the author presents a distributed simulation system for disaster response activities based on the High Level Architecture (HLA). This simulation system focuses on resource management issues including the allocation of scarce response resources to operational areas and it consists of three major components: (1) simulators for the disaster environment, e.g., simulators for damages, casualties and fire spread, (2) simulators for the operations of personnel and technical equipment and (3) some auxiliary simulators. A Multiagent System which models resource allocation tasks within an Emergency Operation Center (EOC) is linked to this simulation. This paper describes the overall architecture of the system and presents some results based on a prototype implementation.

1 INTRODUCTION

Earthquakes like the recent Pakistan earthquake from 2005 or the devastating 1999 Turkey earthquake show the high vulnerability of societies to natural disasters. One of the most challenging areas in this context is the optimization of *local* disaster response activities during the initial three days after an earthquake has hit. During this Search and Rescue (SAR) period major activities include the allocation of available response resources, search and rescue of trapped victims, medical treatment of injured persons, fire fighting, as well as repair of damaged traffic infrastructure and other lifeline systems. Because such events do not happen frequently local decision makers are often not used to process large amounts of incoming information and to include relevant information in their decisions. Regular training methods for field personnel and emergency operation center (EOC) staff members such as table top or full scale exercises help decision makers to be more familiar with such unlikely situations. In addition to those traditional training methods virtual simulation-based training environments are getting more and more important. A variety of off the shelf products are available for emergency response (Jain and McLean 2003), but most available systems are more or less general information, communication and coordination software packages with visualization components, which do not provide active decision support by offering adequate courses of actions dependent on the current situation (Ashcroft et al. 2003).

In this paper the author presents the distributed simulation system EQ-Rescue (EarthQuake-Rescue). This system models the initial response activities after an earthquake disaster. EQ-Rescue includes different simulators which simulate the affected disaster area and the operations of response resources in the field. Simulators are linked through a common architecture for distributed simulations which is provided by the High Level Architecture (HLA). To control the field resources elementary actions are defined. These actions can either be initiated by EOC personnel or by software agents which optimize the allocation of available response resources.

The paper proceeds as follows. After a brief introduction to the used methods (Section 2) a system overview of the model EQ-Rescue is given (Section 3.1). The different components of EQ-Rescue are described in more detail in Sections 3.2 and 3.3. The paper concludes with a description of the prototype implementation in Section 4.

2 METHODS

Simulation systems for disaster response training and response should include simulators for dynamic aspects of the disaster world (e.g., fade away times of trapped persons) and for the response operations of the resources. Additionally it should be possible to link the simulation to computer based decision support systems which may be used by EOC staff during the decision process. Such complex systems can hardly be run on a single personal computer. An adequate architecture for distributed simulations over computer networks is provided by the High Level Architecture. So far HLA-based simulations have been successfully applied in the military domain, but more recently there are increasing efforts to apply HLA to civilian domains, such as traffic and logistics (Strassburger 2001) or emergency management (Klein 2001).

Because simulation systems per se do not offer recommendations on how to act efficiently, an additional decision component must be added to the system. One possible approach, which is also used in EQ-Rescue, is to include autonomous or semi-autonomous software agents which have the capability to reason in dynamic environments.

2.1 The High Level Architecture

HLA was developed by the Defense Modeling and Simulation Office (DMSO) of Department of Defense (DoD) with the main goal of building a platform for war gaming and training taking into account interoperability and reuse of different simulation components. HLA has it's origins in the Distributed Interactive Simulation (DIS) and Aggregate Level Simulation Protocol (ALSP). The development of the HLA started in the mid 90's and since 2000 it is an Institute of Electrical and Electronics Engineers (IEEE) standard for distributed simulation systems (IEEE 2001a, 2001b, 2001c).

A distributed simulation using the HLA is called a federation, and each single simulator in such a federation is referred to as a federate. A federate may be a simulation, live component (e.g., physical device or human operator) or data viewer (Figure 1). Federates may join or resign from a federation at any time during the life cycle of the federation. The HLA provides standardized formats for federation and federates. It consists of general set of HLA Rules, an interface specification, which defines the standard for the Run Time Infrastructure (RTI) and a model object template (OMT). The RTI defined in the interface specification is the central component of an HLA-based simulation. All communication flow between the federates is transferred via this RTI. For a more in depth description of the HLA see for example (Fujimoto 2000, Kuhl et al. 1999) or the standard specification documents (IEEE 2001a, 2001b, 2001c).

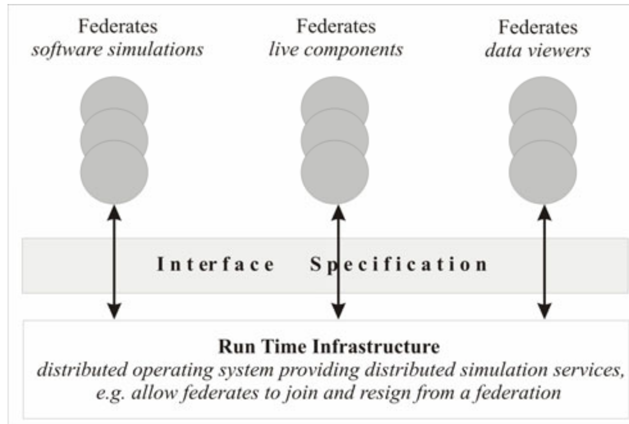

Figure 1: Possible Components of an HLA-based Simulation (after Fujimoto, 2001)

2.2 Multiagent Systems

Because the research area of Multiagent Systems (MAS) is a rather young research field many different definitions and agent classifications exist in literature. Generally speaking a software agent is a software program which acts autonomously or semi-autonomously in a real or simulated environment. The agent senses it's environment and based on the sensor input and possibly internal reasoning mechanisms it initiates actions which change the environment. A MAS is a system where multiple dependent or independent agents act together in the same environment (for an introduction to MAS see for example (Weiss 2000, Wooldridge 2002). One well known agent architecture which is also used for the EQ-Rescue model is the Belief-Desire-Intension (BDI) architecture. The BDI-approach was very much influenced by Bratman's analysis of the human decision making process (Bratman 1987) and software agents following this paradigm have been applied successfully to different real world problems such as fault diagnosis for the space shuttle or air-traffic management (Ingrand et al. 1992). A BDI-agent pursues its given goals (*desires*), adopting appropriate plans (*intentions*) according to its current set of data (*beliefs*) about the state of the world.

3 THE MODEL EQ-RESCUE

3.1 System Overview

Typical EOC functions include a huge variety of different responsibilities, such as declaration of local emergency, disseminating emergency public information, tracking resource allocation, etc. EQ-Rescue is limited to the topic of resource management with the emphasis on prioritizing the allocation of the available resources. Simulators for the disaster environment and for the resource operations are linked together in a central system where agents or decision makers can assign resources to operational areas. Inventory data from the affected area is stored in a central database. This data includes information about the building stock, road network, demographic data, etc. The damage estimation tool EQSIM (Fiedrich et al. 2004) is used to calculate the initial damage scenarios. All system components implement an HLA-interface to enable the use in a common federation. Figure 2 gives an overview of the system architecture.

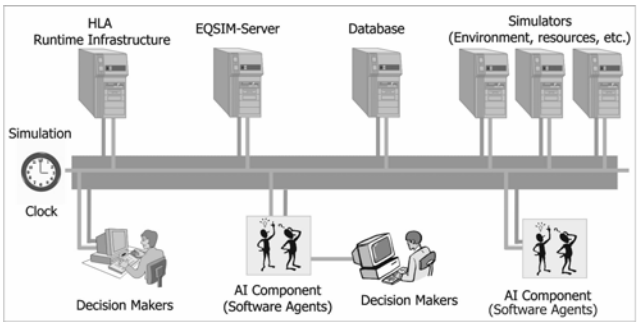

Figure 2: EQ-Rescue Software Architecture

EQ-Rescue consists of three interacting models:

1. A disaster world model which is used for a realistic representation and simulation of the disaster area.
2. A resource model which simulates the movement and the operations of the response resources in the field.
3. An EOC-model which allows EOC members to send orders to and to receive reports from the resources.

Broadly speaking the first model describes how the disaster situation would proceed if no further response activities are initiated. The simulators of the disaster world model have complete knowledge of the actual situation. This includes information about real damage states of buildings and road infrastructure, health states of injured and trapped persons as well as initially ignited buildings. Therefore the world resulting from these simulators can be specified as the Complete Information World (CIW).

Resources in the field have only limited access to the disaster information provided by the disaster world simulators. Each single resource has a sight radius in which it can sense the disaster world in a predefined resolution. The assignment of tasks to the resources can only be initiated by external stimuli such as orders by EOC members or agents. The subsequent operations such as movement and resource work at the assigned areas are simulated within the single resource simulators.

EOC members and agents don't have access to the complete information of the CIW. In fact their decisions are based on incomplete and fuzzy information. They use a second view of the disaster world – namely the Incomplete Information World (IIW). All the knowledge about the IIW comes from the pre-disaster-database and from on-site information of the response resources and teams in the field. The disaster world model and the resource model are the simulation components of EQ-Rescue while the EOC model represents the decision component of the model.
The flow of information between the different components is outlined in Figure 3.

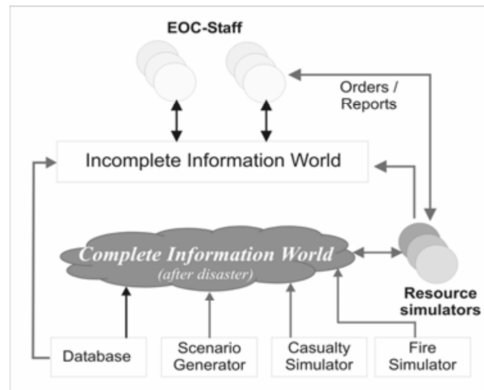

Figure 3: Information Flow in EQ-Rescue

3.2 Simulation Component of EQ-Rescue

The simulation component consists of several federates, where each federate uses a simulator class and an HLA interface class. The simulators are implemented as discrete event simulators using event queues for event scheduling. The interface class uses a time stepped approach to allow the agents to submit messages at any time during the runtime of the federation. The HLA interface is responsible for the coordination of the federate with the overall simulation. The general structure of the an EQ-Rescue federate is shown in Figure 4.

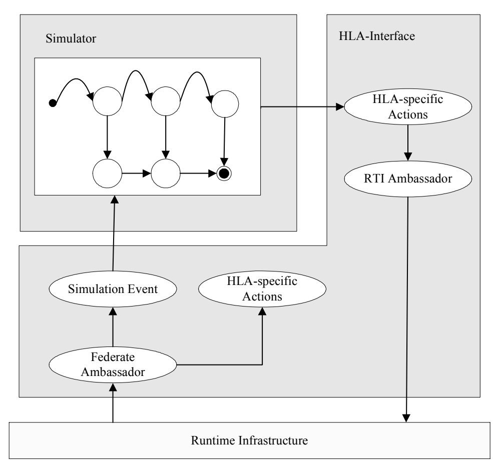

Figure 4: Structure of an HLA Federate in EQ-Rescue

The interface class holds two different queues: (1) an internal queue, which processes the HLA events produced by the simulator class and (2) a callback queue, which processes the HLA callbacks from other simulation and agent federates. The main simulation loop of a federate can be described by the following pseudo code:

```
public void mainSimulation() {
    while (Sim.getTime() < Sim.endTime){
        requestTime = Sim.getTime()+timeStep;
        Sim.increaseTime();
        Sim.simulate();
        while (!internalQueue.isEmpty()) {
            event = internalQueue.deque();
            event.dispatch();
        }
        timeAdvanceRequest (requestTime);
        while (!advanceGranted) {
            cb = callbackQueue.dequeue();
            advanceGranted = cb.dispatch();
        }
    }
}
```

The main loop checks whether the simulator already reached the simulation end time. As long as the simulation is not over, the simulation time is increased and the simulator processes all relevant events and creates relevant HLA messages for other federates. In the next step the interface class sends these messages via the RTI-Ambassador. HLA callbacks from other federates are processed until the RTI grants the advance of the simulation time. Further implementation details about used programming languages and software packages can be found in Section 4.

3.2.1 CIW Federates

The CIW federates simulate the disaster world There are three major federates:

- **Fire Simulator**: The fire simulator uses an approach comparable to (Takai 1999). Three detailed physical models are combined in the fire federate:
  - A combustion model for the fire spread within a burning building,
  - a heat release and ignition model which calculates the possible heat release from a burning building and the ignition of the neighboring buildings and
  - an extinguishing model which simulates firefighting using water.
- **Casualty Simulator**: This simulator models the health state of the injured persons. Therefore the health state is normalized to values between 0 and 100. Each injured person can be assigned to one of four injury classes. The initial injury class is dependent on the location of the person within a collapsed building. Reduction of the health states can be calculated with published survival rates for trapped victims (Coburn et al. 1991). After rescue the reduction rate of an injured person is decreased as soon as they get initial treatment. As soon as an injured person arrives at a hospital it is simply assumed that the health state is stabilized and does not decrease anymore.
- **Damage Estimation Tool EQSIM**: This component serves two purposes: (1) it is used to calculate the initial damage scenario which provides basic input data and (2) the decision makers can benefit from this module and estimate possible damages within the observed area. An HLA-compliant version of EQSIM (Fiedrich et al. 2004) serves this purpose. The underlying methodology considers the building behavior under earthquake load and it is based on the capacity-spectrum-method (ATC, 1996).

3.2.2 Resource Federates

The EQ-Rescue model includes resources which are relevant for the work within the initial Search and Rescue (SAR) period. The main goal of this period is to minimize the number of fatalities and the impact of dynamic secondary disasters. An overview of the relevant resources and their primary tasks is given in Table 1.

Table 1: Resource Simulators and Their Tasks

Resource	Primary operations
Plane / Helicopter	Air reconnaissance
Recon Team	Ground reconnaissance
Specialized SAR-Team	Search and Rescue of trapped persons in collapsed buildings
Search Team	Search of trapped persons in collapsed buildings
Ambulance	Transport of injured persons to hospitals
Crane	Road clearance and support of rescue work
Fire Engine	Firefighting of burning buildings
Dozer, Truck, Excavator, Roller	Road clearance

Resources simulators are modeled with discrete simulation models, including internal resource events as well as external HLA events. The communication flow between resources and agents uses realistic communication protocols based on standard protocols. This communication flow is modeled through HLA interaction classes. The resource

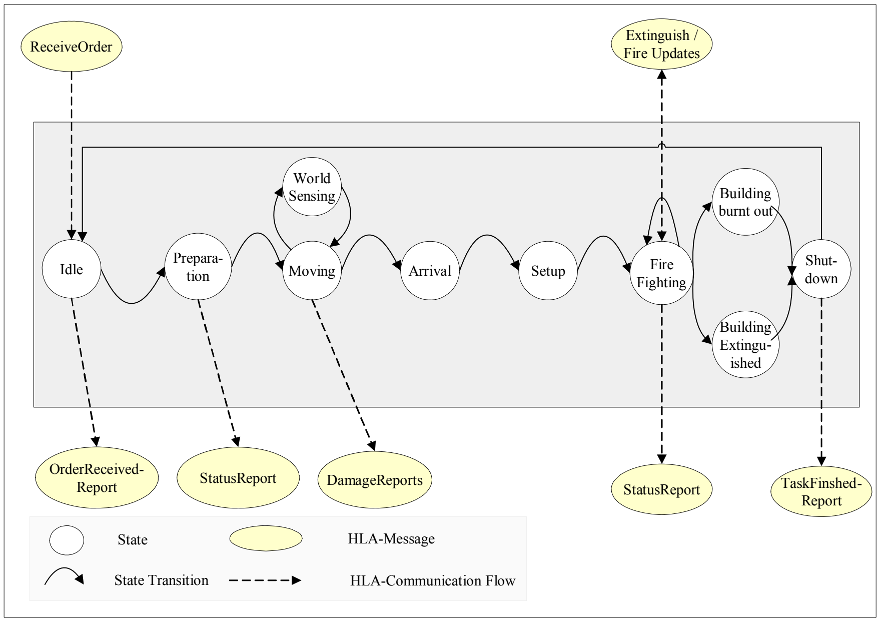

Figure 5: HLA-based Simulation of Fire Brigades

simulators check for incoming HLA-coded orders from the decision component and if an order is received, events for the operational work of the resource are generated and stored in an internal event queue. After the initial preparation time of the resource the resource starts to move on the network to the assigned operational area. During this movement a resource senses its surrounding world and sends HLA-coded reports about the environment. Each resource has a predefined radius within which the world can be sensed with a specified accuracy. The resulting information helps to update the IIW and it can be used within the further decision process. The resource movement on the network is restricted by its width, height and weight. Technical specifications define estimated working times for the simulation. Each resource has a set of elementary work tasks, which can be performed by this resource (for more details, see Fiedrich et al. 2000). An incoming order is split and the associated elementary work tasks are simulated. It is also possible that one task involves more than one resource, e.g., if a crane supports SAR-work at a building. In these cases HLA mechanisms are used to coordinate a single work task over different federates. Figure 5 gives an example of the behavior within the simulation of a single fire engine.

3.2.3 Auxiliary Federates

There are some additional auxiliary federates which are used to run the overall simulation. The first one is a manager federate which coordinates the setup and shutdown process of the federation. The manager federate waits until all relevant simulators have joined the federation and it ensures that all participating federates receive relevant initialization data from other federates. The second auxiliary federate serves as a time pacer and keeps the simulation running in real-time or any other chosen time model. Additionally, all simulation and agent activities are stored in a central database via a database federate. The log generated by this federate can be used for analysis of the agents' and humans' decisions.

3.3 Decision Component of EQ-Rescue

The decision component of EQ-Rescue is modeled through BDI agents. Like the federates described in Section 3.2 all agents implement a HLA interface comparable to Figure 4. This interface allows communication and data exchange between simulators and agents. Therefore all agents can act as federates in the overall simulation. To coordinate their actions it is necessary that agents communicate with each other during the planning process. This communication is not modeled via the HLA. Instead the Agent Communication Language (ACL) is used for inter-agent communication. ACL was developed by the Foundation of Intelligent Agents (FIPA) to enable platform independent agent communication. Further details about ACL can be found in the ACL specification documents (FIPA 2002a, 2002b, 2002c). The agents' behavior is controlled by following types of events:

- Internal Events: These are events which are sent internally within an agent. They are used for the agent's reasoning process.
- External Events: External events occur either through observed changes of the environment or through communication between the agents.

- Motivation Events: These events are generated to create the current goal set of each agent.

Within an EOC the responsibilities are split between different EOC members. Therefore different software agents with different roles are defined. Altogether five different agents with responsibilities for reconnaissance, search and rescue, medical service, fire fighting and road clearance are defined. The plan library of each agent is based on expert knowledge about the agent's respective domain and it is derived from analysis of emergency plans and standard operation procedures as well as from expert surveys. In the planning layer Operations Research (OR) models are included to optimize the assignment of available resources in real-time. The agent's reasoning process follows a layered approach, where – dependent on the incoming messages and the internal reasoning – events are passed to one of the following layers: (1) an action layer, which includes all events which can be handled directly, (2) a planning layer, which includes plans which are solely based on the agent's deliberation and (3) a cooperation layer, which requires co-ordination with other agents. The general agent architecture is shown in Figure 6. A specific evaluation function allows to assess the quality of the agents' decisions. This function includes components for loss of life and property damages due to fires.

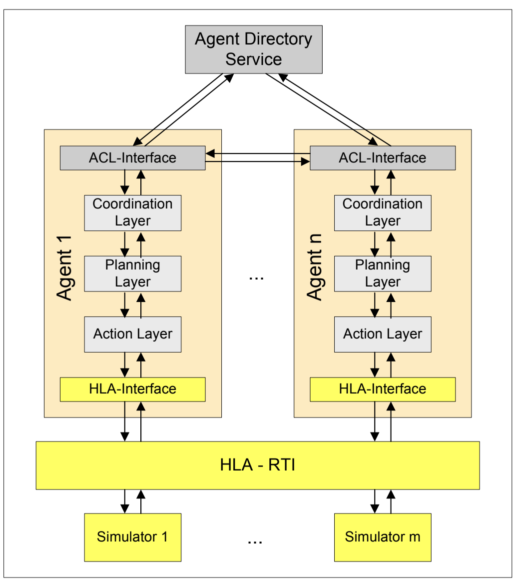

Figure 6: Agent Architecture in EQ-Rescue

4 IMPLEMENTATION AND MODEL TESTING

The implemented prototype of EQ-Rescue includes 18 different federates and five HLA-enabled agents. The simulation federates were implemented in Java. HLA-related tasks, such as development of the HLA object model and the implementation of HLA-specific parts of the federates, BDI-agents. Inter-Agent-Communication was realized with the JACK and FIPA compliant component FIPA-JACK from RMIT University, Australia. Real world data from a defined area in Bucharest and another city in Romania with approximately 700 buildings and 40,000 inhabitants is being used for model testing.

Three different scenarios of varying complexity have been defined and the simulation behavior and the decisions of the agents have been analyzed. The average scenario for the smaller Romanian city included approximately 70 response resources, ten blocked roads, eight initially ignited buildings and 250 seriously injured people which were trapped in several buildings. To run the simulations a small Microsoft Windows network with three personal computers with Pentium 4 processors was used. A simulation of the 72 hour response period took approximately three hours, where the main processing time was used for the agents' reasoning process. Although the simulations were faster than real-time the agents could initiate appropriate actions.

5 CONCLUSIONS

Training and decision support for local emergency managers is essential for successful disaster response. Computer-based simulation and decision support systems may provide a better understanding of the dynamics and complexity which arises after earthquakes and other natural disasters. The emergency management community can benefit from the development of the HLA which was developed for training purposes in the military domain. Nevertheless it is necessary to develop new specific simulation systems which model the dynamics of the emergency response operations. By using the HLA it is possible to build reusable disaster related simulators which might be linked in different disaster related simulations. In this paper the author presented the model EQ-Rescue, which includes a HLA-based simulation of the initial response activities after an earthquake. This model also includes an agent-based component which autonomously optimizes the allocation of the response resources. The presented version of EQ-Rescue does not include an agent-interface to the decision makers. This important component is part of ongoing research. By using an evaluation function for the chosen actions it is possible to quantify decisions of agents and EOC staff. Decisions of agents may therefore be used as a point of reference during training sessions.

ACKNOWLEDGMENTS

Research for this paper has been supported by the Collaborative Research Center 461 *Strong Earthquakes: A Challenge for Geosciences and Civil Engineering*, funded by the German Science Foundation DFG.

REFERENCES

Applied Technology Council (ATC) 1996. *Seismic Evaluation and Retrofit of Concrete Buildings*, Vol. 1. ATC-40, Redwood City (CA), USA.

Ashcroft, J., D.J. Daniels, and S.V. Hart. 2002. Crisis Information Management Software (CIMS) Feature Comparison Report. NIJ Special Report 197065, U.S. Department of Justice, National Institute of Justice, Washington, DC.

Bratman, M.E. 1987. *Intention, Plans, and Practical Reason*. Cambridge, MA, USA: Harvard University Press.

Fiedrich, F., F. Gehbauer, and U. Rickers. 2000: Optimized Resource Allocation for Emergency Response after Earthquake Disasters. *Safety Science* 35 (1-3), 41-57.

Fiedrich, F., J. Leebmann, M. Markus, and C. Schweier. 2004. EQSIM: A New Damage Estimation Tool for Disasters. *Proceedings of the International Conference "Disasters and Society - From Hazard Assessment to Risk Reduction"*, Karlsruhe, Germany, 26 - 27 July.

FIPA 2002a. FIPA Abstract Architecture Specification. Available online via <http://www.fipa.org/specs/fipa00001/SC00001L.pdf> [accessed July, 11, 2006].

FIPA 2002b. FIPA ACL Message Structure Specification. Available online via <http://www.fipa.org/specs/fipa00061/SC00061G.pdf> [accessed July, 11, 2006].

FIPA 2002c. FIPA ACL Message Structure Specification. Available online via <http://www.fipa.org/specs/fipa00026/SC00026H.pdf> [accessed July, 11, 2006].

Fujimoto, R.M. 2000. *Parallel and Distributed Simulation Systems*. Wiley Series on Parallel and Distributed Computing 1. New York, Chichester: John Wiley and Sons.

IEEE 2001a. *IEEE Std 1516-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*. Institute of Electrical and Electronics Engineers, New York. ISBN: 0-7381-2619-5.

IEEE 2001b. *IEEE Std 1516.1-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification*. Institute of Electrical and Electronics Engineers, New York. ISBN: 0-7381-2621-7.

IEEE 2001c. *IEEE Std 1516.2-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification*. Institute of Electrical and Electronics Engineers, New York. ISBN: 0-7381-2623-3.

Jain, S. and C.R. McLean. Modeling and Simulation for Emergency Response. NIST Internal Report NISTIR-7071, National Institute for Standard and Technology, Gaithersburg, MD, USA. Available online via <www.nist.gov/msidlibrary/doc/nistir7071.pdf> [accessed July, 11, 2006].

Ingrand, F.F., M.P. Georgeff, and A.S. Rao. 1992. An Architecture for Real-Time Reasoning and System Control. *IEEE Expert* 7 (9): 33-44.

Klein, U. 2001. *Verteilte Simulation im ausnahmetoleranten städtischen Verkehrsmanagement*. Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, Germany.

Kuhl, F., R. Weatherly, and J. Dahmann. 1999. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Upper Saddle River, NJ: Prentice-Hall.

Strassburger, S. 2001. *Distributed Simulation Based on the High Level Architecture in Civilian Application Domains*. Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, Germany. SCS-Series "Advances in Simulation" AS-11, SCS-Europe BVBA, ISBN 1-56555-218-0.

Takai, H. 1999. Development of the Fire Spread Model for Kobe City. Research Report No. 30, School of Engineering, Kinki University, Higashi-Osaka, Japan.

Weiss, G. (Ed.) 2000. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. 2nd edition. Cambridge, London: The MIT Press.

Wooldridge, M. 2002. *An Introduction to Multiagent Systems*. Chichester, England: John Wiley & Sons.

AUTHOR BIOGRAPHY

FRANK FIEDRICH studied Industrial Engineering with a focus on Computer Science and Operations Research. He earned a Dr. Ing. (Ph.D.) in Civil, Geo and Environmental Engineering at Karlsruhe University, Germany. Currently he is an Assistant Professor at the Institute for Crisis, Disaster, and Risk Management at the George Washington University. He has been working since 1996 in the area of natural disaster research. His main research interests are distributed simulation systems for virtual emergency response training environments and the application of Artificial Intelligence and Information Technology to disaster response. His e-mail address is <fiedrich@gwu.edu> and his personnel web page can be found at <http://www.gwu.edu/~icdrm/People/fiedrich.htm>.