Distributed Agent-Based Building Evacuation Simulator

A. Filippoupolitis, E. Gelenbe, D. Gianni, L. Hey, G. Loukas, S. Timotheou
Intelligent Systems and Networks Group
Imperial College, London SW7 2BT, UK
{afil, e.gelenbe, gianni, l.hey, gl01, stelios.timotheou}@imperial.ac.uk

Keywords: Building Evacuation, Emergency Management, Distributed, JADE, SimJADE.

Abstract

The optimisation of the evacuation of a building plays a fundamental role in emergency situations. The behaviour of individuals, the directions that civilians receive, and the actions of the emergency personnel, will affect the success of the operation. We describe a simulation system that represents the individual, intelligent, and interacting agents that cooperate and compete while evacuating the building. The system also takes into account detailed information about the building and the sensory capabilities that it may contain. Since the level of detail represented in such a simulation can lead to computational needs that grow at least as a polynomial function of the number of the simulated agents, we propose an agent-oriented Distributed Building Evacuation Simulator (DBES). The DBES is integrated with a wireless sensor network which offers a closed loop representation of the evacuation procedure, including the sensed data and the emergency decision making.

1. INTRODUCTION

The final outcome of an emergency situation depends heavily on the management of the actors involved [1]. To minimise emergency evacuation times and to reduce casualties, optimal, fast and decentralised decision making is needed for the direction of the evacuees and the allocation of the emergency personnel. For this reason, new decentralised optimisation techniques based on neural networks [2] have been developed to support the personnel in the decision making process. Agent simulation, which incorporates adaptation and learning mechanisms [3], offers a valuable opportunity to design and test evacuation procedures. To this purpose, a prototype of a one-floor evacuation simulator was initially developed [4]. However, a more realistic scenario of a multi-storey building with a large number of civilians and emergency personnel requires computational resources that are at least of polynomial order in the number of the agents involved. The needed computational resources are usually not available on a single host, and a distributed environment is needed to support such studies. In this paper, we present a *Distributed Building Evacuation Simulator (DBES)* that overcomes such limitations. The DBES has also been integrated with a *Wireless Sensor Network (WSN)*, which provides the remote emergency personnel with the sensed conditions inside the building. Such integration augments the simulator's realism and accuracy in terms of physical and network representation.

The paper is organised as follows: in the related work section, we highlight the differences with other state-of-the-art contributions. Then we introduce the simulation model and how it has been adapted for the distributed execution. We proceed by illustrating the outlines of the simulator, including the SimJADE framework [6] and the integration with a wireless sensor network. We conclude with the validation of the simulator in a simple evacuation scenario.

2. RELATED WORK

Since the events of September 11th 2001, much attention has been directed to the management of emergency situations. Several simulators were developed to allow preliminary studies of such scenarios and training of the emergency personnel. To the best of our knowledge, there have not been any distributed building evacuation simulators before ours. There have been, however, a few significant simulators that can be related to our work.

DrillSim [7] is a multi-agent simulation environment for crisis response. Its main goal is to evaluate new emergency response techniques and to provide a training environment for first responders. However, it cannot operate in a distributed manner. On the contrary, DBES is extended to a distributed environment through the *High Level Architecture*

(HLA) [9], and thus benefits from reduced execution time and increased memory capabilities [10]. Moreover, Drillsim uses a grid based approach for the modelling of the physical world, which is computationally demanding for large areas.

SimSITE [11] and DEFACTO [12] are distributed virtual environments where humans can interact and train in emergency scenarios. DEFACTO deals with an emergency situation at a larger scale, such as a university campus or a block of buildings. The part of the evacuation procedure taking place inside the buildings, however, is not simulated. SimSITE uses a grid for the modelling of the physical world and mainly focuses on the training of emergency personnel through the use of the simulator. These differ from the main goal of the DBES which is the evaluation of optimisation techniques.

3. SIMULATED MODEL

The model is derived in direct analogy with the real physical system, which is composed of independent intelligent actors who individually decide which resources to use. The actors cooperate or compete for the use of a resource according to their internal objectives and to the conditions of the external world. In a typical scenario, the participating actors are the civilians who evacuate the building, the rescuers who collect injured individuals and the firemen who try to extinguish the fire. The simulated model is then based on the agent paradigm [15], and includes human and hazard agents.

The *human agents* are provided with their own personal view of the world, and with their own decision, motion and health models which describe their status. This approach, based on the separation of concerns [16], allows a more accurate modelling of the physical system, while at the same time provides solid foundations for the design of the simulator. The simulation dynamics in terms of pattern of interactions is indeed not affected by the values given by such models.

Hazard agents, such as fire-spreading and smoke-spreading agents, affect the conditions of the simulated world, but do not occupy physical space. They present a simpler simulation dynamic and constitute an independent group.

These models have been adapted to exploit the fundamental parallelism of physical space inherent in a multi-storey building.

3.1. World Model

The world model represents the simulated world within which the agents move and with which they interact. It is composed of a description of the physical space and a description of the world status for each element of the physical world.

A graph is used for the representation of the physical world. The graph nodes model physical Points of Interest (PoI), and the graph edges define the space of the movement within the model. The PoIs represent the physical positions reachable by the agents, which can be of two types:

- physical location of an extinguisher, a desk, a door, etc.
- intersection points between two or more evacuation flows.

The edges define the walking access between two PoIs and are primarily characterised by a length attribute, which indicates the physical world distance between the connecting nodes.

The nodes and the edges are also described by a group ID that uniquely identifies the sub-graph they belong to. Each sub-graph defines a local region in which the change of any attribute of the world is perceived by all the agents located in it. For example, the nodes and the edges modelling a room are part of the same sub-graph because all the agents located there can perceive any changes taking place in it.

In addition, each node can be associated to more than a sub-graph. In such cases the variation of any node of either sub-graph is perceived by the agents standing on the node. Nodes modelling doorways or intersections belong to this category.

The world status adds on top of the spatial plan a set of attributes per each element composing the graph. For each node there is a queue of agents willing to traverse the node, a value for the fire intensity and other such attributes. From the modelling point of view, the node also represents a single server with a FCFS queue attached. It can be occupied by only one human at a time, for a duration exclusively decided by the specific agent. In a normal situation, such number is retrieved through the motion model which we describe below.

Similarly for the edges, lists of the agents crossing each edge are stored, together with the values of its physical conditions. The edges can be considered as infinite servers since they model a segment of the physical space whose length is not negligible and can be occupied by more than one human agent at a time. This modelling simplification reduces the computational requests of the simulator without significantly affecting the results in the case of a scenario with overcrowding. The ability to model the phenomenon of overtaking slower human agents remains.

3.2. Human Agents

The different types of human agents share three models that regulate their movement and their existence in the simulated world. Specific actions and interactions carried out by a single human agent type (for example firemen extinguishing fire or rescuers collecting injured civilians) are directly defined in the simulation logic through the interaction model with the world and with the other agents. The decision, motion and health model are described below.

3.2.1 Decision Model

At each point in time an agent has a goal to achieve. The decision model is the tool used by the agents to take actions towards achieving their goals. It can be seen as a function that maps the goal and the current state of the world to actions.

For example, the goal of the civilian agent is to reach the main exit. Therefore, its decision model provides the next move using the shortest path algorithm towards reaching its destination. However, the weight of each edge considered is not the physical length, but depends on the state of the world.

In initial or in non-hazard conditions, the weights of the edges correspond to their physical length. When the world changes due to fire, or smoke, for example, the decision graph is updated through a customisable updater function that adjusts the edge weights according to the value of the hazard. A simple but effective updater function is the step function that takes a finite value, for fire values smaller than a threshold, and "$+\infty$" otherwise.

The modular design of the model allows the effortless use of a more sophisticated decision model and model updater function. Specifically, the rescuers and firemen might have composite decision models that could use, for example, a neural network [18] to decide what action to take next. For instance, when rescuers have to collect spatially distributed injured civilians, each one of them must decide where to go next in order to ultimately collect all of them in the least possible time. We have to note that the scope of the simulator is to evaluate decision making algorithms and strategies during emergency and time critical situations. The human behaviour aspect, such as experience-based and collective decision making, is not incorporated into the decision model. The structure of the simulator, however, allows for a straightforward integration of various decision making models

3.2.2 Motion Model

This model determines the time duration characteristics of the movement of each agent on the nodes and on the edges. It is defined through the specification of the speed values on both elements as a function of the agent state, the agent characteristics and the physical condition of the node or edge, on which the movement act is occurring.

These values can be constant, as in evacuation trainings, or be function of the perception of danger in the event of real emergencies, and depend on the physical condition of the specific agent.

This model plays a role in the pace of the interaction within the dynamic of the simulator. It determines at what simulation time the events related to the movement acts will be completed.

3.2.3 Health Model

The health model determines the reduction rate of the lifetime of an agent as a function of the local environment (smoke, fire) and of the agent's characteristics (age, personal protection gear).

This model constitutes a barrier to the completion of the movements, since they require a non negligible time, like movement over an edge, and are conditioned by the lifetime given by the model.

3.3. Hazard Agents

In fire engineering and emergency management domains, there are various fire and gas propagation models, which depend on the building's structure [19].

Currently, the simulator includes a fire agent, whose behaviour can either be predetermined, with a manual description through XML configuration files, or probabilistic. In either case, the fire intensity on each node and edge is represented as a number between 0 and 10, and propagates on an extended world graph. This graph inherits the structure of the plan and adds edges between physically adjacent nodes. In real scenarios, for example, the fire may propagate not only through doors and along corridors, which can be traversed by human agents, but also through walls and ceilings.

In our probabilistic model for the spread of a hazard the intensity of the hazard can increase at the nodes themselves independently of their neighbours, or can spread along the edges between nodes. At each simulation second a Bernoulli trial determines whether the intensity at that node increases from itself. The probability of an increase is proportional to the current intensity at the node. Further trials determine whether fire spreads from each of the node's neighbours. For this, each edge is assigned a weight, which determines the rate of spread relative to the other edges. For example, fire may spread more quickly through regular doors than through fire safety doors, and more quickly up through a ceiling than down through a floor (which requires the graph to be directed). The probability of success in the Bernoulli trial for each edge is proportional to the intensity at the neighbouring node. During each simulation second each node's intensity may only increase by one. The fact that the probability of fire spread during each trial is proportional to its intensity at the node and neighbouring nodes means that an increase results in a positive feedback, which in turn means that the total rate of increase rises as the simulation progresses.

3.4. Adaptation to the Distributed Environment

The distributed adaptation of our simulator presents us with two main challenges: how to partition the simulated model over the available computational resources and how to improve the simulator performance through adaptations that do not heavily affect the simulated model.

The model is partitioned in order to exploit the intrinsic parallelism of independent physical subsystems, while meeting the memory constraints on each host and minimising the network workload. For instance, the events happening within a floor or along stairs loosely affect the rest of the system; therefore the simulated world is allocated on independent single-floor and single-stairway simulators, each running on a separate host. In addition, since the stairs constitute critical evacuation paths which are going to be traversed by all the agents escaping the building, they might become overcrowded with the number of agents. In that case, a further partitioning could be necessary in order to meet the memory requirements.

A key factor for the performance of the simulator is the amount of data exchanged between the separate simulators. In order to reduce such data and to reduce the complexity of the graph-based decision model, a PoI that belongs to a remotely simulated world section is represented in a condensed way through a Global Point of Interest (GPoI). Which of the remote PoIs is to be locally represented as GPoI in the local environment is determined by the physical structure of the building and by the personal interest of the specific agent.

While moving from one simulator to another, i.e. when reaching a local exit, Personal PoI (PPoI) dictated by the individual agent interests are carried within the agent state. This information is then locally integrated in the standard world knowledge on the remote destination simulator.

For example, on the federate simulating Floor 2 there is a world model characterised by only PoIs and GPoIs of the graph in Figure 1. The carrying of a PPoI from an agent accessing Floor 2 is dynamically integrated with the world model locally stored.

Such modelling approach reduces the quantity of the exchange data among the simulators and also implements the planning for agent movement.

4. DISTRIBUTED BUILDING EVACUATION SIMULATOR

The DBES is built according to modern software engineering practices that tend to separate the model specifications from the general synchronisation and communication facilities. For this reason, we first designed SimJADE, a distributed agent-oriented simulation framework, and then we implemented the model described above. The state of the simulator, which is required to be consistent at any simulation time, is visualised through local GUIs for each floor and stairs, and for the external point of collection.

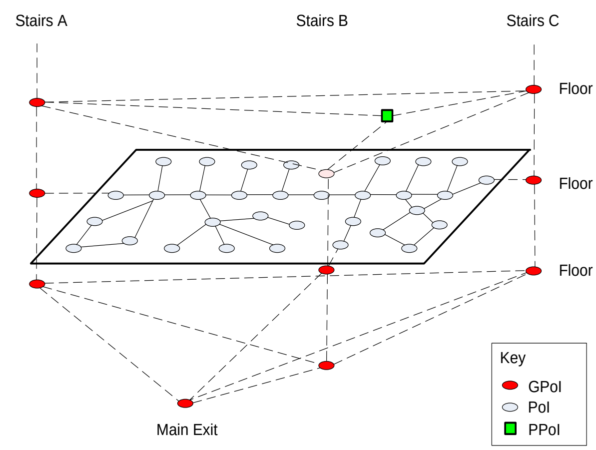

Figure 1 World with Local and Global PoIs

The simulator is also augmented in its realism and accuracy through the integration of a wireless sensor network, which provides real sensed data to the simulated agents. This augmentation makes the simulator useful also during real world emergency situations in buildings provided with sensor networks.

4.1. SimJADE

SimJADE is a simulation framework that extends the popular agent-oriented JADE framework [21] by introducing simulation time based synchronization and communication. Such features, which are introduced on top of an innovative agent-oriented modelling of the DES system, are transparent to the system developers. They can therefore develop the simulator in a very similar way as they would do with the respective conventional MAS [6]. To achieve this, SimJADE defines through JADE's schema: a simulation ontology, a set of simulation agents, a set of simulation behaviours and a set of simulation event handlers.

The simulation ontology, named DES-Ontology, defines the *DES concepts* (simulation time) and *actions* (DES and simulation life cycle management services) that are used as semantic base for the communications among the simulation agents.

The simulation agent society contains a simulation entity agent and a simulation engine agent. The *simulation entity agent* encapsulates the simulation logic, i.e. the sequence of state evolutions and DES service requests, and locally provides discrete event simulation versions of conventional JADE services, such as *doWait* and *receiveMessage*.

The *simulation engine agent*, which may be unique within the society [6], has the role of requests collector and simulation coordinator. It is available in two transparently interchangeable versions; local and distributed. The distributed version is implemented according to the general trend outlined in [22], in close analogy with the framework SimJ [23]. It is based on a HLA-based implementation of layer 1 of the SimArch architecture [24] (Figure 2). Such layer, which is composed of a Federation Manager [25] and a generic-easily configurable federate, allowed the rapid implementation of the simulator and provided synchronisation support for the implementation of simulation-time stamped mobility of the agents between the federates. Please refer to [6] for details concerning SimJADE, and to [24] and [26] for details of SimArch.

The *simulation behaviours* represent the internal routine to be processed in correspondence with any of the actions defined in the DES-Ontology. They also provide the basic container for the encapsulation of JADE standard behaviours in the simulated environment.

The *simulation event handlers* are the conventional routines to be processed within the simulation engine, for each of the service events such as *send event*, *wake up*, for example.

4.2. Agent Dynamics

At the simulation component layer, on top of SimJADE, the dynamics of the agents are defined. They include two main aspects. The first aspect concerns the movement within the world and the second is related to the updates received from it.

The movement is carried out according to the possible positions defined by the simulated world, nodes and edges. The agent initially standing on a node starts its movement behaviour according to the specification of its personal decision and motion models. The behaviour consists of the generation of an arrival event at the destination node after a given simulation time. Upon the arrival, the agent waits in a queue for the authorisation to occupy the node and, once received, it proceeds and stands on the node for a time given by its motion model.

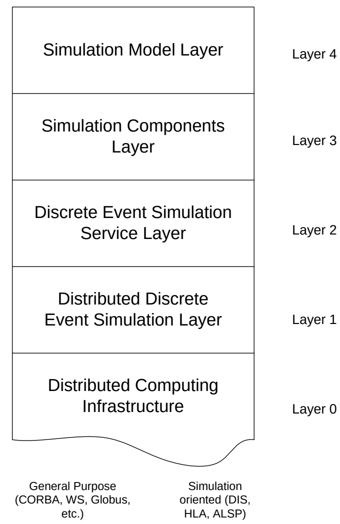

Figure 2 SimArch Architecture [24]

While moving, the agents have to update their personal world model. To reduce the number of events, also considering the rate of the variation of the physical phenomena, the agents receive updates of world changing when they are crossing the edges or when they have just completed their passage through the nodes. The updates are delivered for those parts of the simulated world that belong to the same group of the current position of the agent.

4.3. Graphical Interface

The state of the simulated system can be locally monitored through graphical interfaces that show the local area plan, the position of the civilians and the physical condition on each node or edge. The floor interface is shown in Figure 3, for one of the floors.

The agents' health state and type are shown with different colours. For visualisation requirements, it is fundamental that the system maintains a consistent state at any simulation time instant, which imposes a conservative time management technique.

4.4. Augmented Reality through Wireless Sensor Network Integration

During an emergency, a "smart building" incorporates monitoring and sensing with

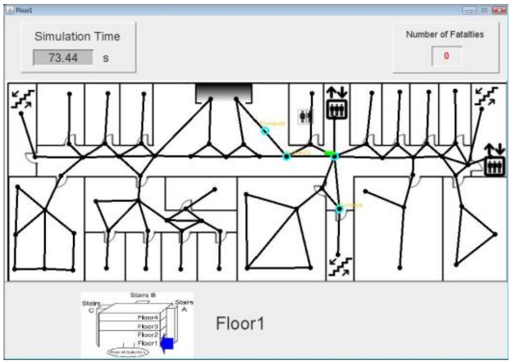

Figure 3 Example GUI – Floor 1

communication and decision making in order to assist both civilians and response personnel. The location of hazards, such as fire or smoke, in the building may be sensed using a WSN. A WSN consists of many low cost motes with simple sensing, processing, and communication capabilities. Its wireless nature reduces infrastructure costs, making its use attractive for such a “smart building” scenario.

Sensed data can be processed in the network if necessary and transferred to a central location for storage or processing. Such distributed monitoring may enable more efficient evacuation of civilians in the building (by avoiding hazards and congestion), and efficient allocation of the emergency response service personnel. In fact, an effective *optimal* decision making needs an as-complete-as-possible knowledge of the scenario in which the actors, civilians and emergency personnel, operate. Part of this knowledge concerns the status of remote locations for which their physical conditions can only be perceived locally by a sensor network and then delivered to the actors. However, to completely assess the impact that a wireless sensor network’s properties, such as limited processing capabilities, losses and delays resulting from multi-hop network routing, have on the scenario, a model of the WSN should be included in the simulator. Unfortunately, such properties are complex and dependent on many environmental factors [29], and therefore difficult to accurately capture in a model. Such approach of reality augmentation has indeed been shown to improve the accuracy of simulation by providing a more realistic representation of the physical and network world [5]. We have therefore extended the BES to include such reality augmentation by integrating with a real WSN.

Our WSN test bed consists of 40 telosb motes [30] each mounted on top of a LED, and which intensity is regulated by the DBES. The simulator regulates the light intensity according to the fire value of the edges and nodes surrounding the mote position in the simulated world. The simulator also updates the knowledge of the remote emergency personnel, who otherwise would not have any data regarding the conditions of the remote area.

5. VALIDATION AND PRELIMINARY RESULTS

The validation of emergency simulators is generally not possible with direct comparison of data from the real world, because emergency metrics, such as total or average individual evacuation time, for a specific building often do not exist until some disaster happens; when they happen the priority is not collecting statistics. However, a preliminary validation can be carried out by properly setting the simulator parameters in a verifiable scenario.

In a public building with a simple structure and populated by employees only, who are familiar with the layout, it is reasonable to assume that the evacuees use the shortest physical path to the main exit. In addition, some simple motion models are available in the fire and civil engineering domain and can be used for such purpose [27].

The validation scenario includes eight simulators plus a Federation Manager. There are four floor simulators each representing a floor of the Electrical and Electronic Engineering Department building, three stairwell simulators and an external point of collection, which registers the evacuation of the civilians. The building is populated with eighty civilians uniformly distributed over the four floors.

The civilians’ motion model is set according to the statistics provided in [28], where people have average speed s of 150 cm/s, and therefore it can be assumed that the time spent on a node, which is about 50 cm, is roughly given by 0.3 second. The edge crossing time is given by l / s, where l is the physical length in cm.

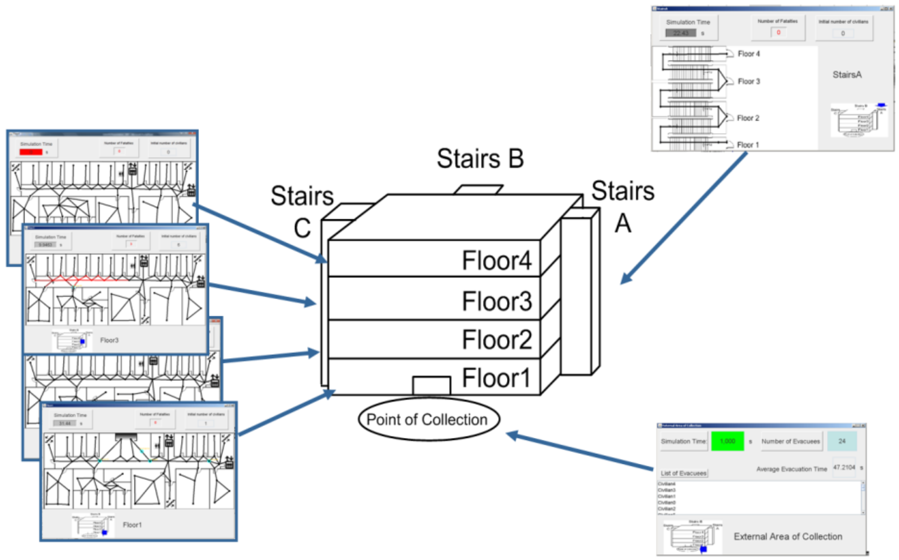

Figure 4 Distributed Building Evacuation Simulator

The metrics collected at the external point of collection are the total evacuation time and the average individual time considering only the movement time.

In our numerical example the results showed that in several runs the total evacuation time is bit less than 87 seconds. We compare this result with the value that can be obtained by applying the mathematical model presented in [28], which gives a minimum total evacuation value of 76 seconds for our scenario. This difference is expected because the mathematical model provides estimations for this metric under ideal conditions.

The average individual evacuation time also contributes to validate the simulator. For the above scenario, in which the population is uniformly distributed over the floors, this metric presents a value that is about the half of the total evacuation time, as expected from the configuration. Other configurations showed that this metric varies according distribution of the people towards the lower floors, as reasonably expected.

6. CONCLUSIONS

In emergency situations, decisions have to be taken quickly and optimally to minimise the evacuation time of the site concerned and the number of casualties. Due to the criticality of such scenarios, a simulator that provides an accurate and detailed representation of the system is needed to design and to evaluate optimal actions. However, in the context of building evacuation, the implementation of an accurate and detailed model of a largely populated scenario requires significant computational resources that are hardly available on single host. In this paper, we present the design of a *Distributed Building Evacuation Simulator*, which effectively allows the simulation of largely populated scenarios. The simulator is integrated with a WSN and provides a general framework within which custom behaviours can be introduced. A preliminary validation of the simulator in a verifiable scenario is also presented.

Further work will include a more extensive validation of the simulator, the incorporation of more realistic human behaviours, and the implementation of decentralised optimisation techniques for the optimisation of the evacuation process.

ACKNOWLDGEMENTS

This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Networks) project and is jointly funded by a BAE Systems and EPSRC (Engineering and Physical Research Council) strategic partnership (EP/C548051/1).

REFERENCES

[1] C.W. Johnson, “Lessons from the Evacuation of the World Trade Centre, September 11th 2001 for the Development of Computer-Based Simulations”, *Journal of Cognition, Technology and Work*, vol. 7, n. 4, Nov, 2005, Springer London, pp. 214 – 240.

[2] E. Gelenbe and S. Timotheou, “Random Neural Networks with Synchronised Interactions”, *Neural Computation*, accepted for publication.

[3] E. Gelenbe, E. Seref, and Z. Xu, "Simulation with Learning Agents", *Proceedings of the IEEE*, vol. 89, n. 2, February, 2001, pp. 148 – 157.

[4] A. Filippoupolitis, L. Hey, G. Loukas, E. Gelenbe, and S. Timotheou, "Emergency Response Simulation Using Wireless Sensor Networks", *The First International Conference on Ambient Media and Systems (Ambi-sys08)*, February, 2008, Quebec City, Canada.

[5] E. Gelenbe, K. Hussain, and V. Kaptan, "Simulating Autonomous Agents in Augmented Reality", *Journal of Systems and Software*, vol. 74, n. 3, February, 2005, pp. 255 – 268.

[6] D. Gianni, "Bringing Discrete Event Simulation Concepts into Multi Agent System", *Proceeding of the 10th International Conference on Computer and Simulation (EuroSim-UKSim08)*.

[7] V. Balasubramanian, D. Massaguer, S. Mehrotra, and N. Venkatasubramanian, "Drillsim: A simulation framework for emergency response drills", *Proceedings of the 2006 Conference on Intelligence and Security Informatics (ISI 2006)*, May, 2006.

[8] Robocup Rescue, http://www.rescuesystem.org /robocuprescue/.

[9] IEEE 1516, Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules.

[10] R.M. Fujimoto, *Parallel and Distributed Simulation Systems*, Wiley, 2000.

[11] Liu, Ke Shen, Xiaojun Georganas, Nicolas D. Saddik, Abdulmotaleb El Boukerche, Azzedine, "SimSITE: The HLA/RTI Based Emergency Preparedness and Response Training Simulation" *Proceeding of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Application (DS-RT2007)*, Oct, 2007, Greece, pp. 59 – 63.

[12] N. Schurr, J. Marecki, M. Tambe, P. Scerri, N. Kasinadhuni, and J.P. Lewis, "The Future of Disaster Response: Humans Working with Multiagent Teams using DEFACTO", Multi-Agent Programming, vol. 15, Springer US, 2006, pp. 197 – 215.

[13] I. Takeuchi, "A Massively Multi-agent Simulation System for Disaster Mitigation", *Proceedings of the Massively Multi Agent System (MMAS2004)*, LNAI 3446, Springer Verlag, 2005. pp. 269 – 282.

[14] F.A. Bowers and D.L. Prochnow, "JTLS-JCATS Federation Support of Emergency Response Training", *Proceedings of the 2003 Winter Simulation Conference*, Dec, 2003, pp. 1052 – 1060.

[15] N.R. Jennings, and M. Wooldridge, "Application of Intelligent Agents", *Agent technology: foundations, applications, and markets*, Springer Verlag, 1998, pp. 3 – 28.

[16] T. Mens and M. Wermelinger, "Separation of concerns for software evolution", *Journal of Software Maintenance*, vol. 14, n. 5, Sept, 2002, pp. 311 – 315.

[17] T. Shen, "Identifying the Target Spaces for Building Evacuation", *Building and Environment*, n. 41, Elsevier, 2006, pp. 1600 – 1606.

[18] E. Gelenbe, "Learning in the Recurrent Random Neural Network", *Neural Computation*, vol. 5, n. 1, 1993, pp. 154 – 164.

[19] British Standard Institute, Fire Safety Engineering in Buildings – Part 1: Guide to the Application of Fire Safety in Engineering Principles, DD240, 1997.

[20] D.G. Elms, A.H. Buchanan, J.W. Dusing, "Modelling the Fire Spread in Buildings", *Fire Technology*, vol. 20, n. 1, 1994, pp. 11 – 19.

[21] F. Bellifemine, G. Caire, and D. Greenwood, "Developing Multi-Agent Systems with JADE", Wiley (2007).

[22] A. Tolk, "Avoiding another Green Elephant – A Proposal for the Next Generation HLA based on the Model Driven Architecture", *2002 Fall Simulation Interoperability Workshop*, 02F-SIW-004, Simulation Standards Organization, Orlando, Florida, September 2002.

[23] A. D'Ambrogio, D. Gianni, and G. Iazeolla, "SimJ: a Framework to Distributed Simulators", *Proceedings of the 2006 Summer Computer Simulation Conference (SCSC06)*, Calgary, Canada, 2006, pp. 149 – 156.

[24] D. Gianni, A. D'Ambrogio, and G. Iazeolla, "A Layered Architecture for the Model-driven Development of Distributed Simulators", *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools08)*, Marseille, France, March, 2008.

[25] F. Kuhl, R. Weatherly and J. Dahmann, *Creating computer simulation systems: an introduction to the high level architecture*, Prentice Hall, 1999.

[26] D. Gianni and A. D'Ambrogio, "A Language to Enable Distributed Simulation of Extended Queueing Networks", *Journal of Computer*, Vol. 2, N. 4, July, 2007, Academy Publisher, pp. 76 – 86.

[27] D.A. Purser, and M. Bensilum, "Quantification of Behaviour for Engineering Design Standards and Escape Time Calculations", *Safety Science*, n. 38, 2001, Pergamon, pp. 157 – 182.

[28] J. Pauls, "Calculating Evacuation Time for Tall Buildings", *Fire Safety Journal*, n. 12, 1987, pp. 213 – 236.

[29] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks", *Proceedings of the 1st international Conference on Embedded Networked Sensor Systems (SENSYS03)*, Los Angeles, California, USA, Nov, 2003.. ACM, New York, NY, pp. 1 – 13.

[30] J. Polastre, R. Szewczyk and D. Culler, "Telos: enabling ultra-low power wireless research", *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, Apr, 2005, pp. 364 – 369.