

Emergency Response Simulation Using Wireless Sensor Networks

Avgoustinos Filippoupolitis, Laurence Hey, Georgios Loukas,
Erol Gelenbe and Stelios Timotheou
Intelligent Systems and Networks Group, Imperial College London
South Kensington Campus, SW7 2BT, UK
{afil, laurence.hey, gl1, e.gelenbe, stelios.timotheou}@imperial.ac.uk

ABSTRACT

During emergency response situations, decisions have to be made in a timely manner. Multiple entities have to be optimally coordinated and numerous resources must be allocated efficiently, creating a very interesting and challenging technical problem. In this paper we present a simulation system that models the evacuation of a multi-storey building. Autonomous intelligent agents are used to represent various types of actors that interact inside a virtual physical world. We also model virtual hazards, such as fire, that spread inside the building evacuation simulator. A real wireless sensor network is used to monitor the spread of the hazards while an external event generator provides input to the sensors. We study the effect of different disaster scenarios and agent behaviours, such as human behaviour during an emergency, on the result of the evacuation procedure. Our initial results indicate that the safety of the evacuees and the evacuation time depend on local interactions between the participants and are affected by the actors' decisions. The integration with the wireless sensor network gives us the opportunity to investigate the effect of sensed information on resource allocation and allows us to study the impact of network issues on the decision making process.

1. INTRODUCTION

Emergency response situations usually involve an incident commander who is responsible for the efficient coordination and allocation of the available resources. However, recent natural and man-made disasters, such as earthquakes, floods and terrorist attacks, have reinforced the need for better emergency response using IT solutions. Information exchange, coordination between the participating entities, allocation of available resources and decision making could benefit from the use of IT solutions.

In this paper we present an augmented reality simulation system that aims at providing real-time decision support during time critical emergency situations. Augmented reality simulation has been shown to provide improved lev-

els of realism in the representation of the physical settings [1]. Our approach combines a multi-agent simulator which is integrated with a real wireless sensor network. A virtual hazard is monitored by the wireless sensor network and the sensed data are used by the simulator.

There are several simulation tools that attempt to study emergency planning and response during a crisis. A potential Sarin gas attack in New York city is studied in [2]. Their goal is to model a large scale urban catastrophe using agents and to provide information to expert policy makers via simulations. DrillSim [3], is a multi-agent simulation environment for crisis response. Its main goal is to evaluate new techniques for crisis response and to provide a training environment for first responders. The Robocup rescue simulator [4] is a multi-agent generic urban disaster simulation environment aiming at providing decision support in emergency situations. The world is modelled as a grid while the disaster site is an urban location with buildings, roads, walkways, people and vehicles. Finally, UsarSim [5] is a simulation tool for robot sensing and teleoperation in a disaster environment. The goal of the authors is to incorporate the results into experimental robots used in rescue and military operations.

We describe the implementation of a simulation environment which deals with disaster scenarios in tall buildings. These are characterised by few available paths and escape routes, several choke points, such as doors and stairs, as well as severely restricted communications, in terms of range and quality. The user of the simulation can modify the model parameters in real-time, evaluating different strategies and approaches inside a dynamically changing environment. According to the classification of evacuation models presented in [7], our model is a hybrid of the behavioural and movement model categories. The occupants of the building exhibit various types of behaviour, while their actions are being performed due to conditions in the building, including congestion and ongoing threats.

The rest of this paper is structured as follows. The modeling approach we followed is presented in Section 2. The implementation details of the simulator are described in Section 3. The results of several application scenarios are presented in Section 4. Section 5 describes the model of the hazard spreading inside the building and the integration with the wireless sensor network. Finally, in Section 6 we present our conclusions and describe our future work.

2. SIMULATION MODEL

The simulation approach we have adopted models therepresentation of the physical world, the hazards it presents, and the movement and decision making of the actors within it.

2.1 The Physical World

The most common approach for modelling the physical environment in a multi-agent disaster response simulation is to use a grid of equal-sized square cells. The smaller the size of each cell the better the accuracy of the representation. However, when dealing with a large multi-storey building this sort of modelling becomes computationally extremely demanding and not necessarily accurate. Thus, we chose to view the physical world of a building as a collection of “Points of Interest” (PoI) and available paths between them, which form a directed graph. When the path between two locations is blocked, this is easily represented by the loss of the corresponding links or a prohibitive increase of their costs of movement. Similar representation of the physical world can be found in [8, 9].

Following this approach we can benefit from several existing algorithms solving graph theory problems. For example, all actors in a building are most of the times expected to follow the shortest path, which is easily obtained by the Dijkstra algorithm, while the problem of finding the optimal path that a rescuer should follow to look for injured civilians is analogous to the travelling salesman problem.

2.1.1 Points of Interest

Each PoI is assigned to a group according to its location, for example, a room or section of a corridor. Connecting these groups of PoI adds the option of using a coarse network representation of the physical world to facilitate the searching algorithms when the complete graph is not needed. We have constant and variable attributes for each PoI. Each PoI has constant attributes that are predefined in a given scenario. They include the unique ID, the geographical $\{X, Y, Z\}$ location, the group the PoI belongs to, the type (door, corridor, room, stairs), and a description text. Variable attributes change according to the course of events, and include the availability of a set of special properties (wireless access, fire extinguisher, etc.), the set of actors that are there, and the degree of danger that the PoI poses to an actor’s life.

2.1.2 Links

The links represent the walking access between adjacent PoI and are also defined by a set of constant and variable attributes. Constant attributes can be the unique ID, source and destination PoI, length, type (corridor, room, stairs), and description text. Variable attributes consist of the degree of risk and condition of the path. The time t that an actor needs to cross a link depends on the link length l , the speed of the actor v , and the path’s condition d , simply by: $t = \frac{l}{vd}$.

2.1.3 Congestion

Each choke point (doors, stairs) is modeled as a single server with one queue. When an actor arrives at a choke point, it either finds it free and starts being served or finds it busy (another actor is being served) and has to wait in the respective queue.

2.2 The Actors

Each actor participating in the simulation has a subjective view of the physical world. So, while it is aware of the correct topology of the graph of PoI, its knowledge of the link costs of the graph, the number and type of actors in each group of PoI, and the risk at each PoI and link, can be outdated or wrongly perceived. Various types of actors can participate in our simulator, each on with different characteristics. For example, civilians who have to reach one of the evacuation exits while they try to minimise their injuries or rescuers who search for injured civilians and lead them to the evacuation exits. Other types of actors that can be modeled are firemen, evacuation wardens and bomb disposal personnel.

2.2.1 Movement of the Actors

The actors’ movements is affected by different attributes associated with their special characteristics and the condition of the environment. We specifically consider three main attributes that affect their movement: the sense of duty towards PoI (D), the risk associated with the particular PoI (R), and the movement of the other actors in the surrounding area (I). The general equation describing the goal of an actor a is given by:

$$G(a, n) = k_D(a)D(a, n) - k_R(a)R(a, n) + k_I(a)I(a, n),$$

$$k_D, k_R, k_I \geq 0, a \in A, n \in N \quad (1)$$

where A is the set of actors and N the set of possible destinations. The coefficients k_D , k_R and k_I represent the different importance that an actor puts on each aspect of decision:

Duty - D. An actor a has duty $D(a, n)$, which defines the attractiveness that it associates with a PoI, n . For example, a civilian’s main goal is to evacuate so it is highly attracted to the exits, whereas a rescuer assigns high attractiveness to the locations of injured civilians.

Risk - R. The perceived risk $R(a, n)$ of actor a towards PoI n depends on the condition of the path to that PoI. Specifically, the value of the risk is the sum of the graph edge weights that correspond to the links belonging to the optimal path from the current position to the destination of the actor. The graph’s edge weights depend on the distance between nodes, the condition of the path and the hazard. Hence, the edge weights are not constant but adapt as the hazard spreads dynamically. For example, a path’s weights can increase due to a fire, a crashed wall or an obstacle.

Imitation - I. Actors of the same type and in the same area influence each other’s behaviour, often producing a grouping behaviour of movement [10]. For this reason we introduce the imitation factor. Each actor a is affected by the value of the goals of its neighbours g towards different PoI. Neighbours of a are considered the actors that belong to its geographical group, $Group(a)$. In addition, the influence of each of its neighbours is different depending on the “leadership skills” of the particular agent, $k_L(g)$. The equation describing the imitation of agent a for destination n is given by:

$$I(a, n) = \sum_{g \in Group(a)} k_L(g)G(g, n) \quad (2)$$

Whenever the actor re-evaluates its next target of move-

ment, it chooses the one which maximises the goal function (Equation 1) that combines the three aforementioned aspects of decision. Having chosen the next destination, the actual path to follow is chosen by the actor by applying the Dijkstra shortest path algorithm, where the cost of each link is the time needed to cross it.

2.2.2 Updating Actors' Knowledge

Each agent has a partial knowledge about the physical world which is often outdated. For example, when a civilian is in a specific room it cannot know about a disaster taking place in another section of the building. An important feature of the simulator is that it allows for each actor to update its view of the physical world. More specifically, whenever an actor arrives at a PoI that belongs to a different PoI group, it updates its knowledge with the correct attribute values of all the PoI and links that belong to the particular group.

3. EVACUATION SIMULATOR

We have constructed a building evacuation simulator which is based on the discrete event simulation method [11]. The core of the simulator is process based [12, 13, 14]. The progress of each entity through its process can be temporarily stopped due to two reasons:

Unconditional Delays These occur when the progress of an entity is delayed for a time period that can be predetermined. This unconditional event can be added to a list called Future Events List(FEL). When the simulation time reaches the execution time of this event, the respective entity is resumed and it continues through its process. For example, the time it takes a civilian to move between two Points of Interest can be predetermined and can be scheduled.

Conditional Delays In this case the progress of an entity through its process is temporarily halted due to a specific condition not being met. The entity remains at this state until the respective condition is satisfied. These types of events are added to a Conditional Events List(CEL). For example, a civilian has to wait in the queue until a door is free and he is the first person in the queue.

3.1 Scheduler

The simulation scheduler is responsible for the advancement of time and the coordination between different processes. The operation of the scheduler is based on three phases:

FEL Scanning The scheduler scans the events in the Future Events List and determines which are to be executed now.

Execution of Due Now Events The unconditionally delayed events that are due now (they can be more than one), are executed and are removed from the FEL.

CEL Scanning The scheduler scans the Conditional Events List and attempts to move each entity further through its process. If the respective conditions are satisfied, the entity moves on and it will generate a new conditional or unconditional event which is added to the appropriate list.

3.2 Physical Topology

As we previously described, the physical topology is represented by a graph. The implementation of the graph is done in JUNG [15] which is an open source Java library for graph modelling. It facilitates the creation of large graphs and al-

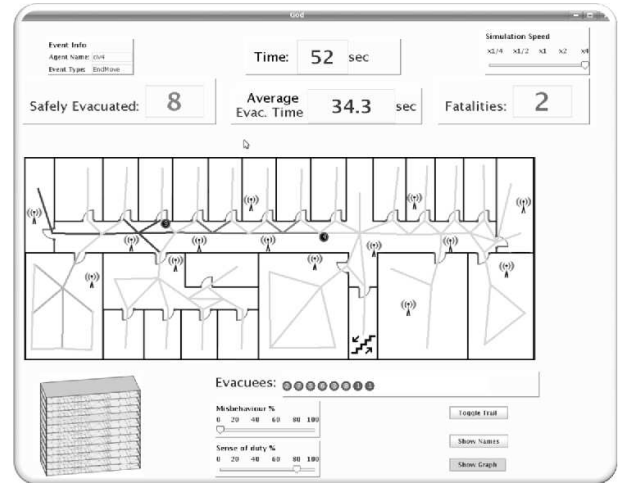


Figure 1: Graphical User Interface of the simulator

lows for annotation of each node and edge with metadata. It also includes the implementations of several algorithms from graph theory.

3.3 Actors

We have represented the actors taking part in the simulation as agents. The implementation of the agents was done using JADE [16]. This is a JAVA framework for agent development that allows for flexible modelling of an agent using behaviours, actions and other useful properties. The coordination of the agents is done by the simulation scheduler. The latter appropriately chooses the next event that has to be executed and gives control to the respective agent. Each time an agent resumes its activity, it evaluates its position and its goal and decides upon its next action. We must also note that there are specific conditions that have to be satisfied in order for the agent to engage in some activities. When the agent has decided on its next action, it generates an appropriate event and sends it to the scheduler.

3.4 Graphical User Interface

The GUI, shown in Figure 1, consists of a main floor view where we can observe the movement of the actors as they travel inside the building and the spreading of the hazard throughout the specified floor. General information about the simulation, such as current event type, current simulation time, number of actors that evacuated the floor and number of fatalities are shown on the respective panels on the visual interface. Furthermore, the user can use the GUI to change parameters of the simulation in real time: the simulation speed and the actors duty coefficient can be modified while the simulation is running providing an intuitive way to experiment with different scenario settings.

4. APPLICATION SCENARIOS

For the modelling and evaluation of the following scenarios, we consider a 10-storey office building equipped with three staircases and a main exit at level 1, as shown in Figure 2.

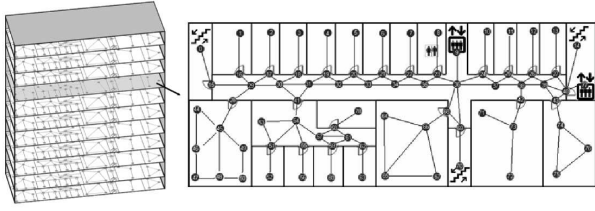


Figure 2: Representation of the seventh floor of the building used in our scenarios.

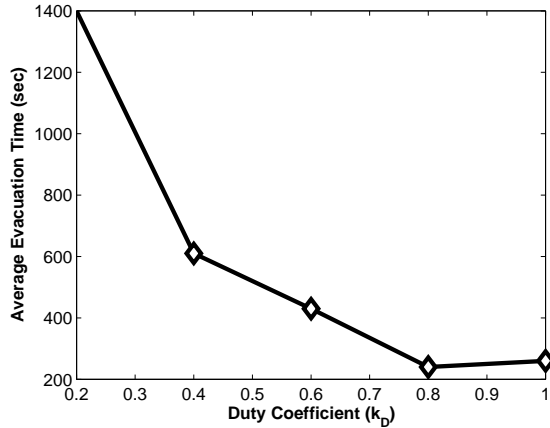


Figure 3: Average evacuation time for varying duty coefficient

4.1 Evacuation of Healthy Civilians

In this scenario Equation 1 is incorporated to model the movement of the actors. It examines the evacuation time of the civilians when the duty coefficient k_D varies and the imitation term is absent. Twenty civilians have to evacuate the seventh floor of a building by using the exit which is located at level 1.

As shown in Figure 3, increasing the duty coefficient parameter k_D , results in significantly smaller evacuation times. High values of duty coefficient are responsible for attraction to points of interest such as the exit. Low values in contrast emphasise the risk parameter, resulting in an attraction to closer “safer” points of interest, and longer evacuation times.

4.2 Evacuation of Civilians in the Presence of Leaders

For the following scenario the civilian actors are separated into two groups, “followers” and “leaders”. During an emergency situation, people are likely to follow someone who is more calm and knowledgeable [10]. By employing the *imitation* term from Equation 1, we are able to recreate this behaviour.

Leaders have a high duty coefficient and low imitation coefficient, whereas followers have low and high duty and imitation coefficients respectively. We investigate the effect the ratio $\frac{k_{L leader}}{k_{L follower}}$ (the ratio of the influence exerted by leaders to that exerted by other followers) has on the mean evacuation time. Calm leaders will evacuate faster than their follower counterparts, as a result of their greater duty coefficient, as we have observed in Section 4.1. Imitation in our

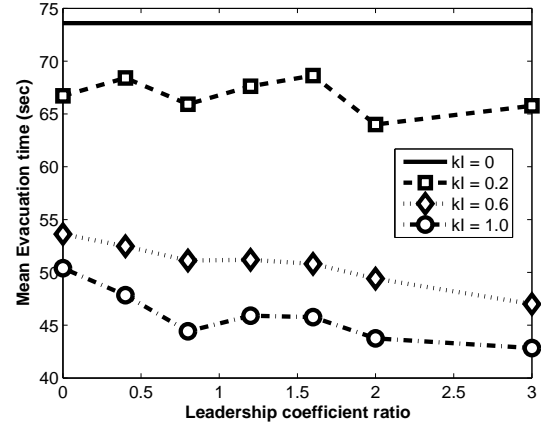


Figure 4: Average evacuation time vs. the leadership coefficient for various degrees of imitation

model results in the followers placing greater influence on the points of interest targeted by others, especially leaders. This should have a feedback effect, with increased interest in a point of interest resulting in further imitation. We would therefore expect groups to form, most probably around leaders, resulting in faster evacuations.

Figure 4 shows the results of ten simulation runs for different leadership coefficient (k_L) ratios, repeated for three values of imitation coefficient. The results are also contrasted with a baseline evacuation time in the absence of imitation. Three leaders join seventeen followers and they all have to evacuate a floor of a building by using the only available exit, as can be seen in Figure 2. We see that regardless of the k_L ratio imitation is beneficial for the evacuation time, with higher imitation coefficients resulting in greater benefit. Furthermore, we observe that if the imitation is high enough, increasing the k_L ratio is also beneficial.

4.3 Effect of Leaders’ Errors

In this scenario we evaluated the effect of leaders’ duty coefficient k_D , on the evacuation time. Each curve depicts the evacuation time versus different values of the leaders’ duty coefficient, while the followers’ duty coefficient is fixed. We also compare each curve with a baseline curve which represents the case where there are no leaders during the evacuation.

Figure 5(a) represents the results for high values of the followers’ duty coefficient ($k_D = 0.8$ and $k_D = 0.6$). Figure 5(b) represents the results for lower values of the followers’ duty coefficient ($k_D = 0.4$ and $k_D = 0.2$). We can note that when the followers have a high value of k_D , the presence of leaders with a lower value of k_D results in a higher evacuation time. Leaders that do not follow their duty, impair the evacuation procedure. The presence of leaders, however, is beneficial in the case where the followers have a low value of k_D . This is shown in Figure 5(b), where we can see that the evacuation time in the presence of leaders is lower compared to the case where the leaders are absent. The intersection between the baseline curve (that shows results without leaders) and the respective curve that corresponds to evacuation in the presence of leaders can help us estimate the point after which the leaders become beneficial for the evacuation.

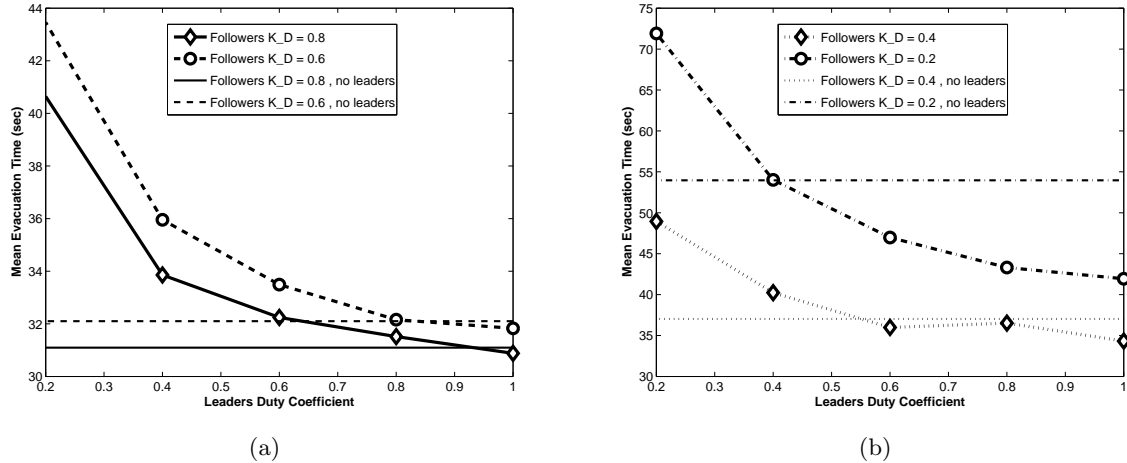


Figure 5: Mean evacuation time vs. leaders' duty coefficient for (a) high and (b) low values of the followers' duty coefficient

5. TOWARDS AUGMENTED REALITY SENSOR NETWORK INTEGRATION

In a building, information on the location and spread of hazards may be relayed to the emergency services and occupants of the building in real time in order to aid addressing issues of tackling the hazards and safely evacuating civilians. We use a real wireless sensor network testbed to monitor the spreading of a virtual hazard, controlled by a separate hazard simulator. These sensed data are sent to and interpreted by the evacuation simulator.

The sensor network testbed consists of 20 sensor nodes, each of which resides within a small black box, which restricts light to the sensors. The light inputs to the sensor nodes are centrally controlled by individually turning four white LEDs on and off which are placed within the tops of each of the boxes. We use light to represent a hazard within the building which the sensor network monitors. The ultimate aim of this architecture is to investigate the impact and benefits that monitoring a hazard with a wireless sensor network will have to the evacuation procedure. For example, the real-time knowledge of a developing hazard could assist the evacuation of civilians and direction of emergency personnel during an emergency procedure. Augmenting the simulation with a real sensor network increases the realism as we capture real network effects such as packet loss and delay. Currently, the hazard is simulated separately and the Building Evacuation Simulator receives the hazard information via the sensor network.

5.1 Modelling the Spread of a Hazard inside a Building

There are examples of graph based models for the spread of hazards such as fire within buildings in the literature. In [17], the authors represent the building as a graph, and use mean fire-resistance as their parameter for modelling the spread of fire between vertices. The model takes into account whether doors are open or not, and the graph is directed (fire spreads up through ceilings faster than it does down through floors). A more complex model for the spread-

ing of the fire as part of a model of building evacuation is presented in [9]. This model places fires within compartments, but does not consider fire intensity, opting instead for time the fire has been burning. Fire spreads between adjacent compartments according to a Bernoulli trial, whose probability of success is defined by a hazard function. Their fires have a duration, after which they do not spread, and cannot be restarted. Furthermore, they list a number of factors which can be modelled using a graph including the spread of fire, smoke, untenable conditions, fire cues, and evacuation of people. The response simulator for which our model is being developed does just that for the evacuation of people and untenable conditions, and in the future will incorporate fire cues.

In our hazard model we use a similar approach. The hazard graph's vertices match those of the Building Evacuation Simulator, but the edges differ. Each vertex v has an integer hazard intensity $H(v)$ between 0 and 9, and the hazard propagates between vertices along connecting edges. Each edge e has a weight $R(e)$ between 0 and 1, which represents the rate at which the hazard might be expected to spread along that edge. We specifically modeled walls, doorways, and sections of corridors represented by the rates r_w , r_d , and r_c respectively.

Depending on the values of the weight parameters used, this simple model can be used to simulate the spread of a hazard through a building. This separate simulation moves in discrete time intervals, and during each interval the hazard intensity at each vertex can only be incremented or decremented. The hazard intensity at a vertex can increase by two means; it can grow independently of neighbouring vertices, or it can spread from a neighbouring vertex along an edge. The probability of the hazard intensity increasing independently is proportional to its current intensity multiplied by a constant, r_f . The probability of it spreading from a neighbouring vertex is proportional to that vertex's intensity multiplied by the rate defined by the connecting edge's constant. We give our three edge types; walls, doors, and corridors different rates dependent on the rate of fire spreading we are attempting to model.

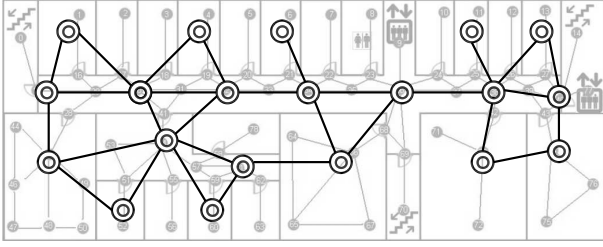


Figure 6: Assignment of sensor nodes to vertices on the simulator graph

5.2 Wireless Sensor Network Integration

Each one of the 20 sensor nodes corresponds to a group of vertices on the graph (e.g. like a room's hazard detector). The brightness of the LEDs is controlled by the intensity of the hazard at the corresponding vertices. Figure 6 shows the hazard model's graph superimposed on the building evacuation simulator. The light levels are periodically read by the sensor nodes and sent to a sink node. Multiple instances of our emergency response simulator can connect to the wireless sink node. They process the gathered data, obtain the sensed representation of the hazard and update the state of their virtual world accordingly. When the intensity of the fire on a vertex increases, the degree of danger for each incident link increases accordingly. Over simulated time the paths become more hazardous and slower to traverse. The actors taking part in the simulation are also affected: when they move on a link with an increased degree of danger, their health level decreases. Excessive exposure to danger results in a fatality.

5.3 Evacuation of Civilians in the Presence of a Hazard

To evaluate the safety of the evacuation process under different hazard conditions we have performed simulations with varying "duty coefficient" k_D .

We present results for two cases: slow and fast spreading fire. For each case we executed ten simulation runs and averaged the results. In each case the simulations for five different duty constants were run in parallel so that the evacuation process would be performed under the same conditions.

Figure 7 shows the mean number of fatalities and injuries for twenty actors. As might be expected, faster spreading rates result in a higher number of fatalities for both hazard spreading rates. We observe that the number of actors fatally injured by the hazards decreases as the duty coefficient increases. This results from the fact that evacuation time decreases for larger values of duty coefficient, as was previously shown in Section 4.1. Higher k_D results in more injuries. This is because, of those that are caught in the hazard that are sustaining injuries, a higher proportion will escape before being fatally injured.

These results are expected from the model. Lower duty, relative to risk, means that actors will favour closer points of interest where the danger is low and will take longer to evacuate. As the attraction to the exit gradually increases, the hazard is spreading, and once the low duty, high risk actors attempt to evacuate he must pass the hazard and will be injured, potentially fatally.

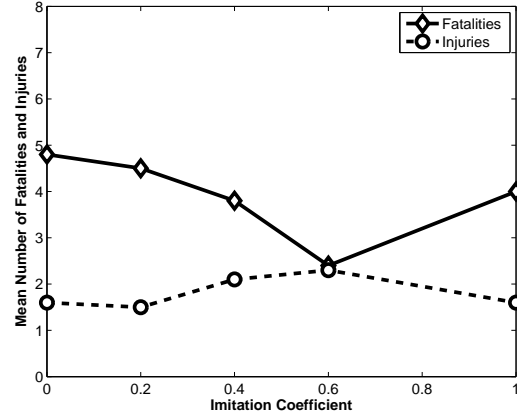


Figure 8: Mean Number of Fatalities and Injuries versus the imitation impact

5.4 Impact of Leadership on Fatalities and Injuries

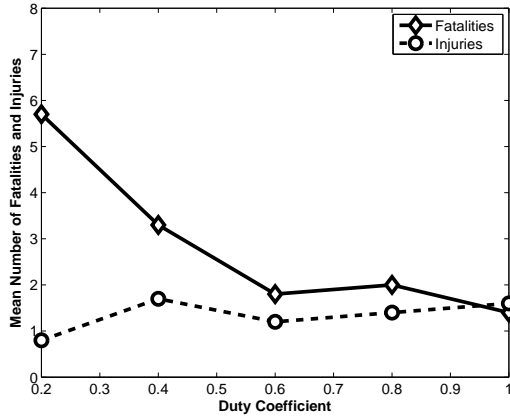
We employ the slow fire scenario to further examine the impact of imitation and leaders on the safe evacuation of civilians. Three leaders, as defined in Section 4.2, are placed amongst seventeen followers. We measure the number of fatalities and injuries sustained for different values of the follower imitation coefficient. The leadership coefficient ratio is fixed at 2.

Figure 8 shows the results. Increasing the impact of imitation results in fewer fatalities. We however observe an unexpected result for an imitation coefficient of 1, in that there is a sharp rise in the number of fatalities. This is a side effect of group forming around a leader whose high duty coefficient results in an attraction to a point of interest which becomes untenable as a result of the spreading hazard. Rather than the single fatality which would result in the absence of imitation, an entire group is affected. We can therefore see that there is an optimum value of the imitation coefficient, which results in the lowest number of fatalities.

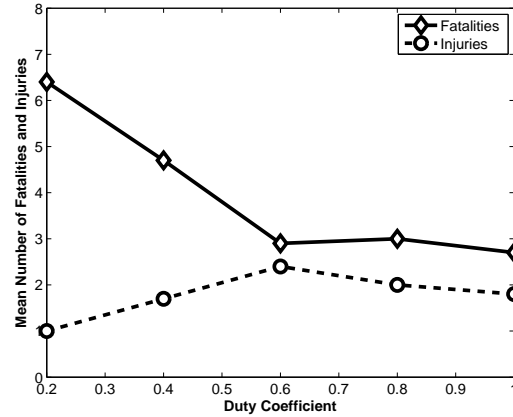
6. CONCLUSIONS AND FUTURE WORK

The augmented reality simulation system that we present can be operated during a disaster to evaluate evacuation strategies in real-time. It can also evaluate evacuation policies for a specific building and examine whether it abides by the relevant standards, by simulating the spreading of a threat in it or receiving real data through integrated sensor networks. We have shown the effect of individual and collective behaviours and the influence that local agent interactions have during an evacuation procedure, including grouping behaviours and the impact of leadership.

The Building Evacuation Simulator can directly process the events of the hazard simulator and obtain the actual representation of the hazard inside the building. By using two separate representations of the virtual hazard (sensed and actual), we are able to investigate the effect of sensed information on resource allocation and decision making. We can, for example, supply firemen with information about the areas where a hazard is present. Moreover, we have the opportunity to study the effect of network issues and incom-



(a) Slow fire



(b) Fast fire

Figure 7: Mean fatalities and injuries resulting from (a)slow and (b)fast spreading hazard, for varying duty coefficient

plete information (e.g. lost or delayed packets, corrupted sensor measurements) on the decision making process.

We have already operated the simulator in a distributed parallelised manner, with each floor being simulated on a different machine. In future work we will study the use of Random Neural Networks [18] and goal based learning [19] in the behavioural model of the agents. Finally, the simulator will be a testbed to investigate and evaluate both localised and distributed optimisation techniques that can be used to achieve optimal performance.

7. ACKNOWLEDGEMENTS

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

8. REFERENCES

- [1] Gelenbe, E., Hussain, K., Kaptan, V.: Simulating autonomous agents in augmented reality. *Journal of Systems and Software* **74**(3) (February 2005) 255–268
- [2] Mysore, V., Narzisi, G., Mishra, B.: Agent modelling on a sarin attack in Manhattan. In: AAMAS '06. (May 2006)
- [3] Balasubramanian, V., Massaguer, D., Mehrotra, S., Venkatasubramanian, N.: Drillsim: A simulation framework for emergency response drills. In: Intelligence and Security Informatics (ISI 2006). (May 2006)
- [4] : Robocup- Rescue simulator website <http://www.rescuesystem.org/robocuprescue/>.
- [5] Wang, J., Lewis, M., Gennari, J.: A game engine based simulation of the NIST urban search and rescue arenas. In: Winter Simulation Conference (WSC '03). (December 2003) 1039–1045
- [6] Wang, J., Lewis, M., Scerri, P.: Cooperating robots for search and rescue. In: Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06)
- [7] Kuligowski, E., Peacock, R.: Review of building evacuation models. Technical report, NIST Technical Note 1471 (July 2005)
- [8] Ling, W., Williamson, R.: Modeling of fire spread through probabilistic networks. *Fire Safety Journal* **9** (1985) 287–300
- [9] Hasofer, A.M., Odigie, D.O.: Stochastic modelling for occupant safety in a building fire. *Fire Safety Journal* **36** (2001) 269–289
- [10] Gershon, R.R.M.: The world trade center evacuation study: Lessons for other high rise office buildings. In: NFPA World Safety Conference & Exposition. (June 2006)
- [11] Pidd, M.: Object orientation, discrete simulation and the three-phase approach. *Journal of the Operational Research Society* **46** (1995) 362–374
- [12] Gilbert, G.N.: Simulation for the social scientist. Open University Press (1999)
- [13] Bennett, B.S.: Simulation fundamentals. Prentice Hall (1995)
- [14] Banks, J.: Discrete-event system simulation. 3 edn. Prentice Hall (2001)
- [15] : Java universal network/graph framework (JUNG) <http://jung.sourceforge.net>.
- [16] : Java agent development framework (JADE) <http://jade.tilab.com>.
- [17] Elms, D.G., Buchanan, A.H., Dusing, J.W.: Modeling fire spread in buildings. *Fire Technology* **20**(1) (1984) 11–19
- [18] Gelenbe, E.: Learning in the recurrent random neural network. *Neural Computation* **5**(1) (1993) 154–164
- [19] Gelenbe, E., Seref, E., Xu, Z.: Simulation with learning agents. *Proceedings of the IEEE* **89**(2) (Feb 2001) 148–157