An Emergency Response System for Intelligent Buildings

Avgoustinos Filippoupolitis and Erol Gelenbe

Abstract Finding the best evacuation path during an emergency situation inside a building is a challenging task, due to the dynamically changing conditions and the strict time constraints. Information systems can benefit the evacuation process by providing directions to the evacuees in an efficient and timely manner. In this paper we propose the use of such a system and evaluate it with a specialised software platform that we have developed for simulation of disasters in buildings. The system provides movement decision support to evacuees by directing them through the less hazardous routes to an exit. It is composed of a network of Decision Nodes and sensor nodes, positioned in specific locations inside the building. The recommendations of the Decision Nodes are computed in a distributed manner, at each of the Decision Nodes, which then communicate them to evacuees or rescue personnel located in their vicinity. The system computes the best evacuation routes in real-time, while a hazard is spreading inside the building. It also takes into account the spatial characteristics of hazard propagation inside a confined space. Our simulation results show that the outcome of the evacuation procedure is improved by the use of the decision support system.

1 Introduction

An evacuation that takes place inside a confined space, such as a building, is a complex situation. The occupants have to quickly decide which path to

Avgoustinos Filippoupolitis
Department of Electrical & Electronic Engineering, Imperial College London,SW7 2BT
e-mail: afil@imperial.ac.uk

Erol Gelenbe
Department of Electrical & Electronic Engineering, Imperial College London, SW7 2BT
e-mail: e.gelenbe@imperial.ac.uk

follow in order to exit the building safely. This, however, is not an easy task, especially if there is an ongoing hazard present. In this case, conditions can change rapidly as the state of paths may deteriorate with time. Let us assume, for example, an evacuee located on the third floor of a building. The fire alarm goes off and he decides to exit the building through one of the emergency stairways. As he walks down the stairs, he reaches the first floor and starts to see smoke. At this point he realises that the fire has blocked the evacuation path he has chosen. He goes back and tries to find another evacuation path, hoping that it will not be affected by the hazard. However, he has already lost valuable time walking along a blocked evacuation path, which may mean that the fire has spread even further. This is one of the numerous examples that we can use to demonstrate how challenging the evacuation of a building is.

There are various approaches regarding the problem of movement decision support during emergency situations. In [1] the authors propose a distributed algorithm for robot navigation using a sensor network. They evaluate their approach using a robot and a sensor network composed of nine nodes. They do not, however, take into account other parameters such as a dynamically spreading hazard or high number of evacuees, which play an important role in providing decision support during a disaster. The authors in [3] propose an algorithm inspired by sensor network routing, in order to guide a flying robot. Although they also evaluated their method for guiding humans, the evaluation scenario included only one human and twelve sensors positioned inside a building. Scenarios with larger building occupancies and dynamically changing conditions were not investigated. In [15] a system based on sensor networks is proposed, for navigating the user to a goal location by avoiding hazardous areas. The path calculation algorithm is based on artificial potential fields. A testbed of 50 wireless nodes was used to evaluate the approach. The focus of the system was on the time needed by the nodes to obtain the shortest path. The authors, however, did not include an evaluation scenario with a spreading hazard or a large number of evacuees. A distributed navigation algorithm geared towards emergency situations is presented in [16]. The approach is inspired by an ad-hoc network routing protocol and uses hop-count as the distance metric. The authors tested their method using simulation, in various network topologies and hazardous locations. Although the algorithm was able to find the path towards the exits, avoiding hazardous areas, the spreading of the hazard and the presence of evacuees inside the area were not taken into account.

Our goal is to design a system that operates during an emergency situation inside a building. The system should adapt to the changes of the environment and provide directions to the evacuees regarding the best available exit, in real time. We present a fully distributed system that we have designed and evaluate its performance in an evacuation scenario inside a three storey building, using a multi-agent simulation platform that we developed.The remaining of this paper is structured as follows: in Section 2 we present an overview of the simulation platform that we used for the evaluation of our proposed system. Then, in Section 3 we continue with the modelling approach we followed for the design of the system and describe the fully distributed decision support algorithm. The results of our simulations are given in Section 4. We conclude with a summary of our contributions and a description of our future work.

2 The Distributed Building Evacuation Simulator

We have implemented the proposed decision support system and the respective algorithm inside the Distributed Building Evacuation Simulator (DBES) [8, 5]. Our simulation environment is based on the JADE platform, which is a software platform for developing applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems [9]. Our simulation platform relies on discrete event simulation techniques. All entities register to the controlling simulator and define the time at which they are to be awaken. The simulator undertakes the re-organising of the entities and triggers each agent to execute at its corresponding time. Furthermore, it is able to operate in real-time, which facilitates the integration of external components, such as a real sensor network.

The disaster area is modelled as a graph or a collection of graphs, which contain special nodes, such as entrances and staircases. For example, Figure 1 shows two areas with five collection points and three graph bridges. Each of these areas is controlled by a dedicated simulator, and simulator is aware only of its area of interest and how it is connected with other areas via the graph bridges. Similarly, each simulated entity has an initial perspective of the overall areas with a rough estimate of the edge lengths. As these agents move in the graph and interact both with the simulators but also with other agents, this perspective is updated with more accurate data. Using this graph-based approach, we are able to focus on different areas of the simulation.

The simulated actors are agents with individual characteristics. For instance, we can simulate a number of heterogeneous entities such as evacuating civilians, robots that explore the building, injured civilians, rescuers etc. The state of each entity is represented by its location, health level and an individual goal, which is typically the target location of its movement. As the simulation evolves, these parameters can be affected by the environment or other agents. For example, a fire spreading in a building will affect the health of some of the civilians and will block exits. Also, each entity has its own world perspective. This perspective is an initial estimate of the overall graph model which reflects the whole area under simulation. As the simulation evolves and the entities traverse the graph towards the exit or any other node, they update their internal perspective with the current surroundings.

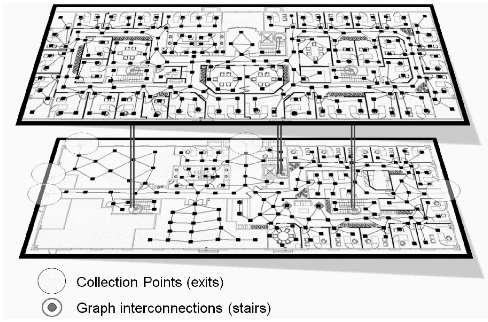


Fig. 1 Two areas with 5 collection points (exits) and 3 graph bridges (staircases)

3 The Decision Support System

The design of our decision support system is based on assumptions closely related to the nature of the environment it is meant to operate in. Our first assumption is the knowledge of the building's layout. Since the system is going to be deployed inside an office building or other multi-storey constructions, it is logical to assume that we have access to data regarding the building's internal structure. Another assumption we make is that a number of Decision Nodes (DNs) are installed in specific locations inside the building. These devices do not need to have high processing power or storage. The role of a DN is to compute the best direction towards a building exit and communicate this to the evacuees present in its vicinity. This can be achieved via a dynamic panel displaying the direction that should be followed, or via a wireless device, such as a PDA, which is carried by the evacuees and receives the direction information from the DNs. Finally, we assume there is a network of sensor nodes installed in the building. Their role is to provide real-time information to the DNs regarding the conditions inside the building, such as the presence of fire or smoke.

The known building layout is used to create a graph G. Each vertex of G represents a location where people can congregate, such as rooms, corridors and doorways. A link between two vertices of the graph corresponds to a physical path that can be followed by the evacuees. The length $l(i, j)$ of a link (i, j) is the actual distance of the path between two neighbouring DNs. Each of the wireless sensors is associated to a link (i, j) and measures the intensity of the hazard $H(i, j)$ along the link. When there is not a hazard present, $H(i, j) = 1$ and its value increases along with the value of the observed hazard. Let us now define the *effective length* of a link as: $L(i, j) = l(i, j) \cdot H(i, j)$. This metric expresses how hazardous a link is for a civilian that will traverse it. When there is no hazard along the link, $L \equiv l$ and the effective length becomes equivalent to the physical length of the link. As the value of H increases, the corresponding link becomes more hazardous to traverse. A

DN is positioned at each of the vertices of the graph G, while a sensor node monitors the hazard intensity along a link between two DNs.

The above definition of the effective length has been initially used in our decision support system [7]. It takes into account only the hazard present along the specific link that a sensor is monitoring and the value of $L(i, j)$ will change only when the hazard value increases along this link. We can, however, take advantage of the unique characteristics of hazard spreading inside a confined space in order to improve the performance of our system. Our approach is inspired by the fact that a hazard, such as fire, that is present in a building location will affect neighbouring locations as time progresses [6, 13]. When a sensor reports hazard value only for the link on which it is assigned, hazard values that are present in nearby locations will not be taken into account. This results in the calculation of paths that may direct the evacuees towards locations where hazard is present nearby. As the hazard spreads inside the building, it will eventually affect the evacuation path and it is only then that the evacuees will be given new directions. This phenomenon can cause an increase in the evacuation time, since the evacuees will have to head to a new destination. Moreover, their health will be affected as they will come in contact with hazardous path sections.

By incorporating spatial hazard information in the effective length, the algorithm can proactively exclude paths that involve travelling near hazardous areas. To achieve this, we let each sensor communicate with its neighbours and incorporate their readings into the "spatial" hazard value H_{sp} it reports. The number of neighbours with which a sensor can communicate is defined by a radius R. Figure 2 depicts an example topology where a sensor is allowed to communicate with its neighbours.

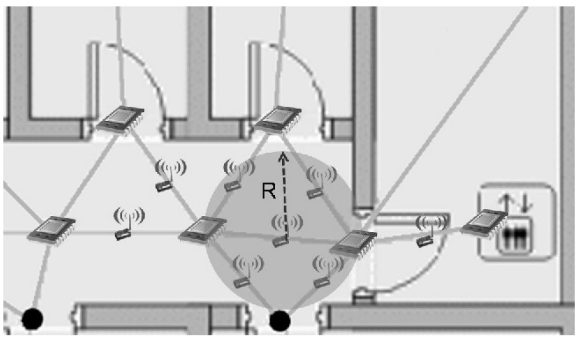


Fig. 2 The sensors included in the neighbourhood area defined by R

Let us now give a new definition for the effective length of a link (i, j), $L_{sp}(i, j)$, which will include the spatial hazard information. Let m be a sensor measuring the hazard level H_m on link (i, j). A sensor n measuring the hazard level H_n on a link (i', j'), belongs to the neighbours set $N(m)$ of m, if:

$d(m,n) \leq R$, where $d(m,n)$ is the Euclidean distance of the sensors locations and R defines the radius of the neighbourhood area. The effective length $L_{sp}(i,j)$ that includes the spatial hazard information is given by $Lsp(i,j) = l(i,j) \cdot H_{sp}(i,j)$, where $H_{sp}(i,j) = H(i,j) + \frac{1}{|N(m)|} \sum_{k \in N(m)} H_k$.

We can now describe the distributed decision support algorithm that will be used in our system. It is similar to principles developed in [12, 10], and inspired by the distributed shortest path algorithm [2, 14, 4] and adaptive routing techniques such as the Cognitive Packet Network [11]. It is executed by each DN, in a distributed manner, and its output is the next DN that is on the best available path towards an exit.

A DN, at vertex u, stores the following information:

- The effective length L_{sp} of all the links that are incident to u
- For every neighbour n of u, the effective length of the path from n to an exit e : $L_{sp}(n,e)$
- The effective length of the shortest path from u to an exit e: $L_{sp}(u,e)$
- The next suggested DN

The initial value for $L(u,e)$ is set to zero if node u is an exit, otherwise it is set to infinity. It is not necessary for a DN to keep information regarding the effective length L of the paths towards all the available exits. As the algorithm is executed, this information is propagated from the exits to the DNs. Each DN will eventually select the exit that minimises the value of the effective length of the path from the node to the exit. The selection of an exit depends on the location of the DN, the locations of the exits and the spreading of the hazard.

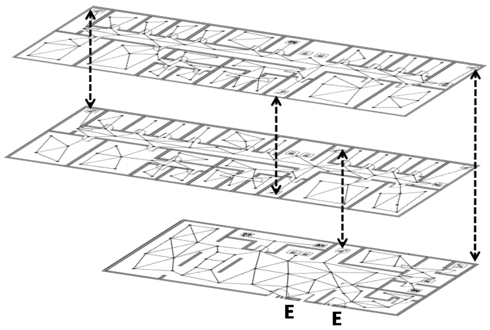
Algorithm 1 Distributed calculation for the effective length $L_{sp}(u,e)$

Send to every neighbour n of u, the effective length of the path from u to the exit e : $L_{sp}(u,e)$
for each sensor node monitoring a link incident to u **do**
    Request hazard intensity H_{sp} from sensor node
    Calculate the effective length $L_{sp}(u,n)$,where n is a neighbour of u
end for
Update the effective length $L_{sp}(u,e)$ of the shortest path to the exit:
$L_{sp}(u,e) = \min \{L_{sp}(u,n)+L_{sp}(n,e): \forall \text{ neighbours } n \text{ of } u\}$
Set the next suggested Decision Node v:
$v = \text{argmin } \{L_{sp}(u,n) + L_{sp}(n,e): \forall \text{ neighbours } n \text{ of } u\}$

When the decision support system is in operation, each DN at u periodically executes Algorithm 1 and provides a suggestion to the evacuees that are in its vicinity. The suggestion is of the form **"go to v"**, where v is a neighbour of u.

4 Simulation Results

In this section we evaluate the proposed decision support system using the DBES. We present simulation results for an emergency scenario that takes place inside a three storey building, illustrated in Figure 3. There are three stair cases inside the building (represented by the dashed lines) which provide access to the different floors, and two exits located on the ground floor. A fire starts spreading on the second floor of the building. The occupancy of the building is 60 civilians (twenty civilians per floor). The sensor neighbourhood area radius is 2m.



(a)

Fig. 3 The building used in the emergency scenario

We have evaluated different cases were the decision system is used, by executing two hundred simulation runs for each case. The difference between them is how frequently each DN executes the algorithm. This will affect the speed of propagation of information among the DNs and the adaptivity of the system to the dynamic environment. We have also included a case where there is no decision support system in the building. Between successive simulation runs, we have chosen to randomise a number of scenario parameters, so that we could test our decision support system in various conditions. The civilians' initial locations on each of the floors, are chosen from a uniform distribution over the respective graph vertices. This randomisation helps us evaluate the performance of the system under different building occupancy patterns. Moreover, the spreading rate of the hazard is different between consecutive simulation runs, since it is based on the probabilistic hazard model used by the DBES [8, 6]. This allows us to test the effectiveness of the system under various hazard spreading speeds.

When we use the decision support system during the evacuation procedure, the civilians move according to the directions of the DNs. Figure 4 depicts the

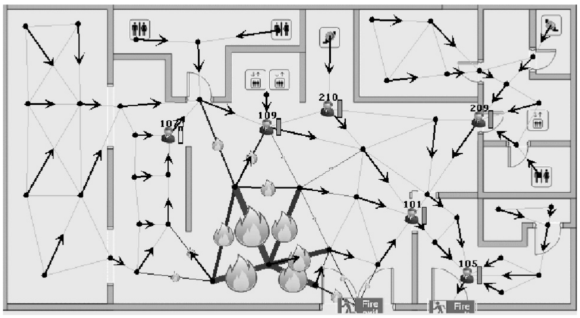


Fig. 4 The decision support system operating inside the DBES. The suggestions of the DNs are indicated by the arrows

ground floor of the building through the DBES graphical interface, where the fire has spread in different locations inside the floor. Each arrow represents the recommendation of the respective DN. The DNs are able to communicate these recommendations to the evacuees by either a visual indicator (such as a smart panel) or a wireless communication device (such as a PDA) which is carried by the evacuees. When the decision support system is not used during the evacuation, we assume that each evacuee has an initial knowledge of the complete building structure. This means that he is familiar with all the building exits and with the corresponding paths that lead to them. This behaviour is implemented inside the DBES by allowing each evacuee to have access to the complete building graph and calculate shortest paths on the graph using Dijkstra's algorithm. When he reaches a location that is affected by the fire, he updates the corresponding graph edge weights and recalculates the shortest path towards an exit.

The percentage of fatally injured evacuees versus the simulation time is a metric that can allow us to determine the efficiency of our system in preventing the exposure of evacuees to the hazard. In Figures 5(a) and 5(b) we can firstly verify that the use of spatial information lowers the percentage of fatally injured evacuees. Furthermore, we must note that the time window ending at the instant of the last fatally injured evacuee is now larger. This effectively means that the evacuees are given directions that lead them away from hazardous areas for a longer time period. In other words, they are able to avoid contact with the hazard for as long as possible. This aspect of the system's behaviour can prove very useful in an operation where emergency personnel (such as rescuers and firemen) enter the building in order to find trapped civilians and lead them to the exit. By keeping the civilians healthy for a longer period of time, the probability of being rescued by emergency personnel that has entered the building increases significantly. Figures 5(c) and 5(d) also illustrate that the use of spatial hazard information minimises the exposure of evacuees to the hazard and decreases the number of fatally injured civilians. Finally, we should note that the execution frequency of the

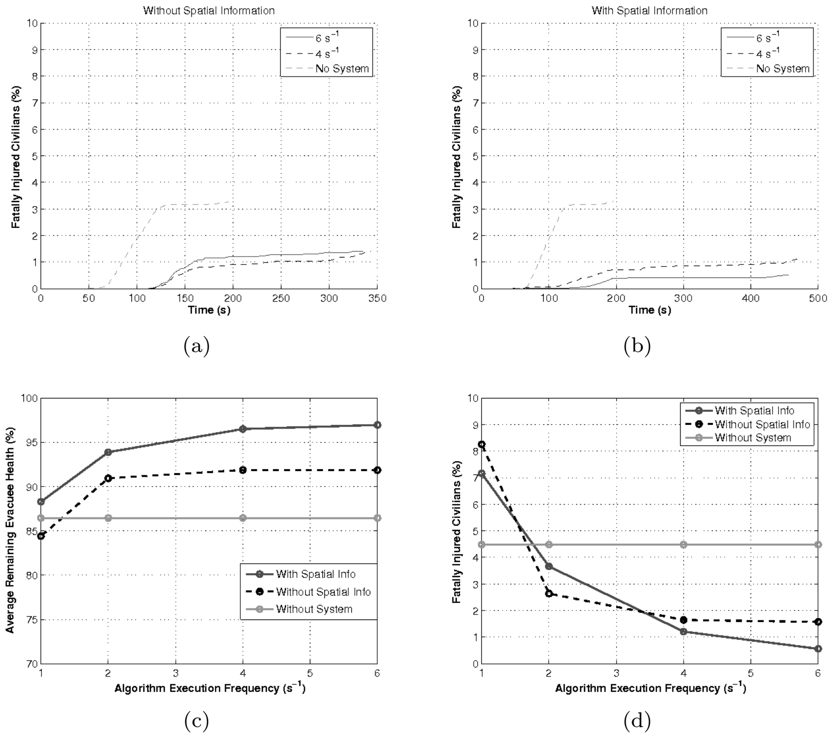


Fig. 5 Simulation results for the evacuation scenario

distributed algorithm affects the performance of the system. This is due the fact that the propagation of the changes in the environment (e.g. the change on the measurement of a sensor node) depend on the execution frequency.

5 Conclusions

We have proposed the use of a system that can facilitate emergency response operations during an ongoing crisis in a building. It consists of Decision Nodes (DNs) that are positioned at specific locations inside the building and sensor nodes that provide information related to the hazard. Each DN uses only local information to compute the best direction towards the exit in a distributed manner and communicates its result to the evacuees via smart panel indicators or wireless devices. The simulation results illustrate that the decision support system improves the outcome of the evacuation procedure by directing the evacuees along safer paths. Moreover, the use of spatial hazard information further improves the performance of the distributed system. In future work we will investigate possible extensions to our system, in order to take into account congestion levels inside the building and uncertainty caused by corrupted sensor readings or defective sensors.

Acknowledgements 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 (UK Engineering and Physical Research Council) strategic partnership (EP/C548051/1).

References

1. Batalin, M., Sukhatme, G.S., Hattig, M.: Mobile robot navigation using a sensor network. In: IEEE International Conference on Robotics and Automation, pp. 636–642. New Orleans, Louisiana (2004)
2. Bertsekas, D., Gallager, R.: Data networks. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1987)
3. Corke, P., Peterson, R., Rus, D.: Networked robots: Flying robot navigation using a sensor net. In: 11th International Symposium of Robotics Research (ISRR 2003), pp. 234–243. Siena, Italy (2003)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Second Edition. The MIT Press (2001)
5. Dimakis, N., Filippoupolitis, A., Gelenbe, E.: Distributed building evacuation simulator for smart emergency management. The Computer Journal **53**(9), 1384–1400 (2010)
6. Elms, D.G., Buchanan, A.H., Dusing, J.W.: Modeling fire spread in buildings. Fire Technology **20**(1), 11–19 (1984)
7. Filippoupolitis, A., Gelenbe, E.: A distributed decision support system for building evacuation. In: Proceedings of the 2nd IEEE International Conference on Human System Interaction, pp. 323–330. Catania, Italy (2009)
8. Filippoupolitis, A., Hey, L., Loukas, G., Gelenbe, E., Timotheou, S.: Emergency response simulation using wireless sensor networks. In: Proceedings of the 1st International Conference on Ambient Media and Systems, pp. 1–8. Quebec City, Canada (2008)
9. FIPA: The Foundation for Intelligent Physical Agents. at http://www.fipa.org
10. Gelenbe, E.: Sensible decisions based on QoS. Computational Management Science **1**(1), 1–14 (2003)
11. Gelenbe, E., Lent, R., Xu, Z.: Measurement and performance of a cognitive packet network. Journal of Computer Networks **37**(6), 691–701 (2001)
12. Gelenbe, E., Seref, E., Xu, Z.: Simulation with Learning Agents. Proceedings of the IEEE **89**(2), 148–157 (2001)
13. Hasofer, A.M., Odigie, D.O.: Stochastic modelling for occupant safety in a building fire. Fire Safety Journal **36**, 269–289 (2001)
14. Humblet, P.A.: Another adaptive distributed shortest path algorithm. IEEE Transactions on Communications **39**, 995–1003 (1991)
15. Li, Q., Rosa, M.D., Rus, D.: Distributed algorithms for guiding navigation across a sensor network. In: MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking, pp. 313–325. San Diego, CA, USA (2003)
16. Tseng, Y.C., Pan, M.S., Tsai, Y.Y.: Wireless sensor networks for emergency navigation. Computer **39**(7), 55–62 (2006)