Simulation Modelling Practice and Theory 16 (2008) 1415–1429

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Crowd simulation for emergency response using BDI agents based on immersive virtual reality

Ameya Shendarkar [b], Karthik Vasudevan [a], Seungho Lee [a], Young-Jun Son [a,*]

[a] Systems and Industrial Engineering, The University of Arizona, Tucson, AZ 85721-0020, USA
[b] Computer Science, The University of Arizona, Tucson, AZ, USA

ARTICLE INFO

Article history:
Received 28 December 2006
Received in revised form 18 July 2008
Accepted 22 July 2008
Available online 5 August 2008

Keywords:
Agent-based simulation
CAVE
Shortest path algorithm
Crowd simulation
Emergency response management

ABSTRACT

This paper presents a novel methodology involving a Virtual Reality (VR)-based Belief, Desire, and Intention (BDI) software agent to construct crowd simulation and demonstrates the use of the same for crowd evacuation management under terrorist bomb attacks in public areas. The proposed BDI agent framework allows modeling of human behavior with a high degree of fidelity. The realistic attributes that govern the BDI characteristics of the agent are reverse-engineered by conducting human-in-the-loop experiments in the VR-based Cave Automatic Virtual Environment (CAVE). To enhance generality and interoperability of the proposed crowd simulation modeling scheme, input data models have been developed to define environment attributes (e.g., maps, demographics, evacuation management parameters). The validity of the proposed data models are tested with two different evacuation scenarios. Finally, experiments are conducted to demonstrate the effect of various crowd evacuation management parameters on the key performance indicators in the evacuation scenario such as crowd evacuation rate and densities. The results reveal that constructed simulation can be used as an effective emergency management tool.

1. Introduction

Emergency response management for man-made and natural incidents has become a key research field in today's world following frequent terror attacks and natural disasters on various crowded areas across the world. The paper focuses on modeling crowd behavior under terrorist bomb attacks in public areas and the required evacuation management. Effective crowd evacuation strategies require accurate prediction of the impact of environment on the behavior of the crowd. Naturally, the involvement of human lives demands high accuracy of such predictions. For these purposes, simulation is an ideal technique as it can accommodate randomness and detail needed in such models, and it enables a form of experimentation not possible with the real incidents. An accurate simulation model can enable the responsible governmental and law-enforcement agencies to evaluate different evacuation and damage control policies beforehand, which can in turn facilitate the execution of the most effective crowd evacuation scheme during the actual situation involving terror. Furthermore, it can allow the training of responders and emergency managers at a fraction of the cost of live training exercises [6].

Simulations involving dense crowds in large cities require a model of an environment and the people in the crowd that are present in that environment. Obtaining realistic data to model such crowds is a challenge in itself. The data so obtained can be so diverse that analyzing it effectively is extremely difficult. Therefore, most of the currently available crowd

* Corresponding author. Tel.: +1 520 626 9530; fax: +1 520 621 6555.
E-mail address: son@sie.arizona.edu (Y.-J. Son).

1569-190X/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2008.07.004

simulations model the crowd as groups of people with common characteristics or objectives. In this paper, we seek to construct a model of an individual with unique characteristics and instantiate it with different attribute values to create a crowd. Since the model of an individual is analogous to an agent, we propose to use an agent-based simulation modeling paradigm to construct a crowd simulation. In this paper, we employ extended BDI (belief, desire, intention) agent framework, which facilitates modeling of a human's mental functions. However, it must also be noted that there is an obvious trade-off between the improved accuracy obtained by introducing the rich BDI framework and computational speed. We plan to tackle this issue as part of our future research work.

The paper is divided as follows: Section 2 summarizes the literature survey and provides brief background information on the various techniques employed in this work. Section 3 discusses an overview of the proposed crowd simulation modeling scheme. Section 4 explains data extraction from safe human-in-the-loop experiments using VR for human behavior to model the agent. Various modeling techniques that facilitate the development of this crowd simulation are discussed in Section 5. In addition, this section also explains the algorithms, data structures and software tools used for the simulation development. Section 6 contains the results from the two phase experimental setup. Phase 1 involves experiments conducted in VR for the purpose of human behavior extraction. Phase 2 involves experiments of crowd simulations to estimate crowd trends and statistics as a consequence of a bomb explosion in the Washington DC mall area, and finally Section 7 concludes the paper and proposes future extensions.

2. Background information and literature survey

This section gives the background information and literature survey of three major areas involved in this paper, namely virtual reality, BDI agents, and crowd simulation.

2.1. Virtual reality

Virtual reality employs detailed computer graphics to create quasi-real 3D objects that respond to user interactions. Three essential characteristics of a VR system are (1) response to human interaction, (2) real time 3D graphics [12] and (3) immersiveness. The first two characteristics are self-explanatory. Immersion means the sense that either the user's point of view or some part of the user's body is contained within the computer-generated space [3]. Immersive virtual reality is defined as the use of various computer graphic systems in combination with various display and interface devices to provide the effect of immersion in an interactive 3D computer generated environment in which the 3D objects have spatial presence. Immersiveness in desktop VR is associated only with the immersiveness of the eye and interactivity is through the mouse or keyboard.

Virtual reality can be of various types, such as Cab Simulators, Projected Reality, Augmented Reality, Tele-presence and Desktop VR. Different kinds of VR have different levels of interactivity and immersions. Most virtual reality used in the engineering applications involves Desktop VR. However, Desktop VR does not completely immerse/involve humans and would result in comparatively insincere participation from the human's part. Hence, in this work, we use the CAVE Automatic Virtual Environment (CAVE) to create the computer generated space or environment. The immersive effect is created using the stereoscopic glasses to create the illusion of 3D in the brain and the 3 wall–1 floor space (to project the environment) to allow the user to stand within the environment itself. The interaction is through a 5-button VR WAND (similar to a joystick) controller and through a tracking system which tracks the position of the wand and the stereoscopic glasses. A high level of realism and response to the interaction is obtained by using a multi-processor high performance Linux computer cluster to run the VR graphics in CAVE. The z buffer and multiple processors ensure high graphic quality and smooth graphic transition. Design process for virtual reality applications has two driving requirements: First, the virtual environment and its interface should be tailored to the task. Second, stringent performance constraints must be met for the benefit of Virtual Reality to be realized [3]. In this work human characteristics for modeling an agent were extracted from VR-based human-in-the-loop experiments, which were designed (using powerful hardware) to meet the afore-mentioned design requirements.

2.2. BDI agents

Modeling human behavior for a given situation is extremely difficult. One of the commonly used ways is to design an intelligent agent, which mimics the overall (abstract) characteristics of a human. The intelligent agent [10] is a promising technique to model a human acting as a decision maker in an automated system. An intelligent agent is situated in an environment and acts autonomously within it, and people are the archetype for autonomous action [4]. An intelligent agent has its own characteristics such as autonomy, social ability, reactivity, and it is pro-active, cooperative, inclined to learn and adaptive [8]. Naturally, the accuracy of such intelligent agents is highly dependent on the modeling accuracy of the behavior of an agent. An agent modeled with the help of a richer architecture will better simulate the behavior of a human.

Various architectures/techniques such as Hierarchical Agent Control (HAC), Cognitive Agent Architecture (Cougaar) and Distributed Environment Centered Agent Framework (DECAF) have been proposed to model such intelligent agents [2].

Among several such techniques, the belief-desire-intention (BDI) modeling is preferred due to it being a widely used, extensible, non-domain specific, flexible and powerful modeling technique for representing human characteristics. A BDI agent is characterized by its "mental state" with three major components: beliefs, desires, and intentions [14]. Beliefs correspond to information the agent has about the world. It may be incomplete or incorrect. Desires represent states of affairs that the agent would wish to be brought about. Intentions represent desires that the agent has committed to achieve. The BDI architecture offers the following main advantages over HAC, Cougaar, DECAF and other architectures:

- First, the BDI paradigm is based on folk psychology, where the core concepts of the paradigm map easily to the language people use to describe their reasoning and actions in everyday life [10].
- The BDI paradigm is a relatively mature framework and has been successfully used in a number of medium-to-large scale software systems [7,13,15].
- The development of such agents is well supported by commercial simulation software packages such as AnyLogic and Jack.

Several researchers have enhanced the BDI framework and applied it to various applications. Rao and Georgeff [13] integrated the theoretical foundations of BDI agents from both a quantitative decision-theoretical perspective and a symbolic reasoning perspective, and discussed the practical application problems. Later, Kinny et al. [7] presented a methodology and modeling technique to describe the external and internal perspective of multi-agent systems, and illustrated the approach for an air-traffic management system. Norling [11] used JACK (commercial software) to encode the knowledge elicitation methodology that mapped closely to the philosophical underpinnings of BDI. Zhao and Son [15] proposed an enhanced BDI framework and applied it for a monitoring agent to detect errors in a complex automated manufacturing system. In this work, we employ the enhanced BDI framework proposed by Zhao and Son [15], whose details and implementation for our specific application (emergency evacuation) are discussed in Section 5.1.

2.3. Crowd simulation

Conventional crowd simulations are based on an assumption that the behavior of crowds emerges from simple rules configured by parameters which are homogeneous and assigned a priori to each individual. Actually, this simple simulation framework has succeeded in representing the behavior of a swarm of fish or birds. However, as modeling humans involves the interaction of various features of psychological and physical attributes, their behaviors have more heterogeneous features as compared with animals that behave more homogeneously [5]. The above described intra-individual interaction (between psychological and physical properties) is interlaced with inter-individual interaction in the simulation to create a more realistic crowd behavior in response to trigger events (e.g., bombing). The agent-based modeling paradigm compared with the traditional discrete event simulation paradigm (where some global processes passively drive entities) has three advantages: First, it does not require knowledge about the global interdependencies, which is required for the discrete event paradigm (top-down modeling). Instead, we define the behavior of an active entity at the individual level (bottom-up modeling), and the global behavior emerges because of many individuals. Therefore, maintenance and modification of the model pertain to the local level and does not require changes at the global level. Second, it allows us to capture more complex structures and dynamics. Third, an agent can have several characteristics allowing modeling of realistic human behavior such as autonomy, social ability, reactivity, pro-activeness, cooperativeness, learning ability, and adaptivity [8]. As mentioned earlier, we use the BDI agent framework to create the agents in the model (see Section 2.2). Each agent has his/her own physical and psychological characteristics and is replicated to create the crowd.

In this paper, we develop a simulation model based on the two-layer modeling principles proposed by Hamagami and Hirata [5]: (1) modeling the agent and (2) modeling the environment that agents interact with. The use of two conceptual layers allows us to isolate modeling of the environment and agent. The interaction between the layers is analogous to the interaction between humans and their surroundings in the real world. The agent makes decisions based on the perceptions from the environment and executes decisions to achieve its goal in the environment. Another aspect of such simulations for emergency response exercises is that they must be run on ordinary office computers and cannot involve expensive licensing agreements to enable local first responders to use these simulations on a day-to-day basis [9]. The simulation developed as part of this work is executed on a workstation PC in AnyLogic software (without involving expensive add-on libraries such as pedestrian library). Other software such as Logo®, Arena®, and JACK® were considered for agent-based modeling and simulation and were rejected owing to their comparative drawbacks. AnyLogic® edges out Logo® in ease of use along with simulation engine power, Arena® in its capability to provide agent-based libraries, and JACK in its simulation scalability features. Further, AnyLogic has a built-in library called AgentBase that facilitates the development of agent-based models and allows the developer to enhance the existing models by allowing Java extensions. AnyLogic supports state-chart-based modeling. which facilitates the modeling of human behavior. The models developed in AnyLogic can be extended to make use of advanced features using more sophisticated add-in libraries (e.g., "Pedestrian" library that facilitates the development of pedestrian behaviors). The developed simulation is also capable of generating emergency scenarios by varying factors such as police distributions, bombing intensities and crowd population with ease.

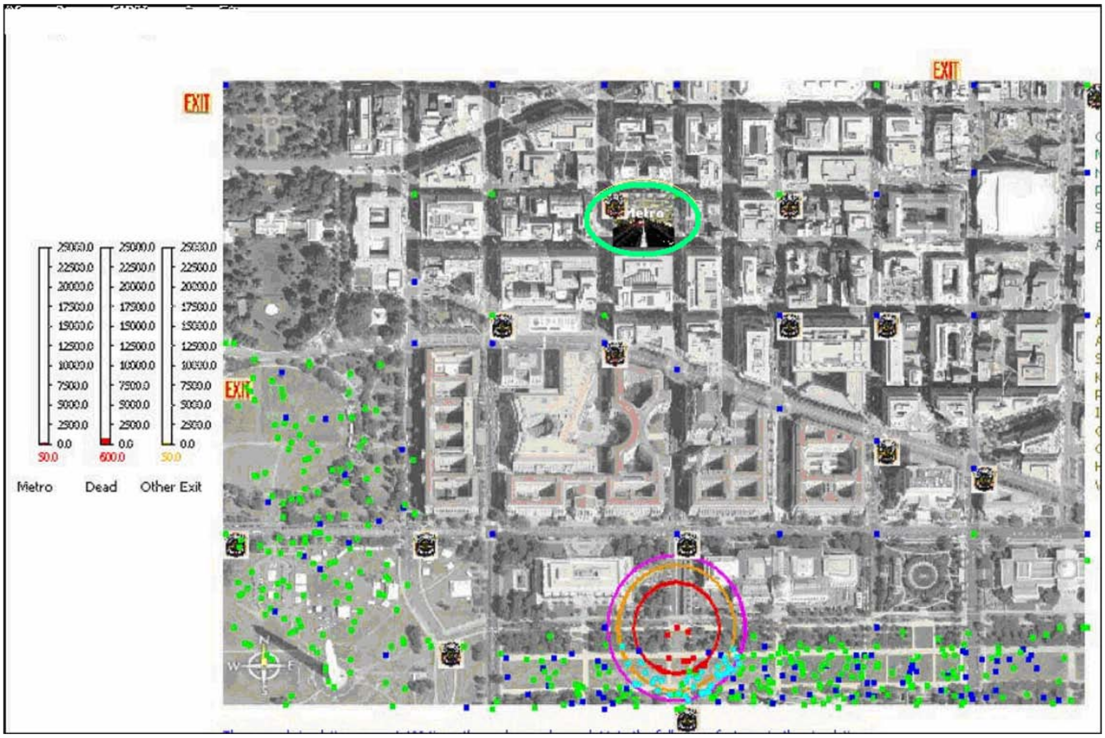

Fig. 1. Snapshot of crowd simulation at National mall, Washington DC.

3. Emergency scenario and simulation development workflow

3.1. Emergency scenario

In this work, we consider the evacuation scenario of a crowd in the densely populated area of the National Mall in Washington DC when an explosion occurs (circled in Fig. 1 that is a snapshot of the developed crowd simulation). Initially in our simulation, people with various goals (e.g., business, shopping, tourist, etc.) are distributed throughout the city. The area also has policemen who know the area and know where exists are located. After the explosion, people try to exit as quickly as possible from the exit points. Depending on various characteristics of people (e.g., knowledge of area, leadership), their evacuation behavior will be different. For example, those who are familiar with the area become group leaders and follow the shortest path to the exit. For those who are not familiar with the area, they move from intersection to intersection and may interact with a policeman or a group leader who could guide them to the nearest exit. Once they reach an exit point, they are removed from the simulation. One example of an exit point is a Metro station which is located well beyond the radius of the explosion and circled at the top in Fig. 1[1] in green. After the explosion, some policemen (first responders) react and approach the area where the explosion took place and guide people out of that area and to the closest exit point to hasten the process of evacuation.

3.2. Overview of simulation development workflow

This section provides a sequential overview of steps involved in crowd simulation development workflow (see Fig. 2) that simulates the scenario mentioned above. The first step consists of data collection which drives perceptions and decisions of an agent that mimics a human under terrorist attack. The VR CAVE is used to set up an immersive environment to conduct human-in-the-loop experiments to collect the required data. The agent data model driven by this collected data is then used to build the state charts to define agent (human) thought processes. The agent's thought process mimics the human behavior and is modeled using the extended BDI architecture. Once an agent model is built, it is replicated (each agent having unique characteristics) to create multiple agents to form the crowd. These agents are then placed with their initial intents into the environment as defined by the input data. Agents then derive their beliefs from this environment and start participating in the simulation.

[1] For the interpretation of color in this figure, the reader is referred to the Web version of this article.

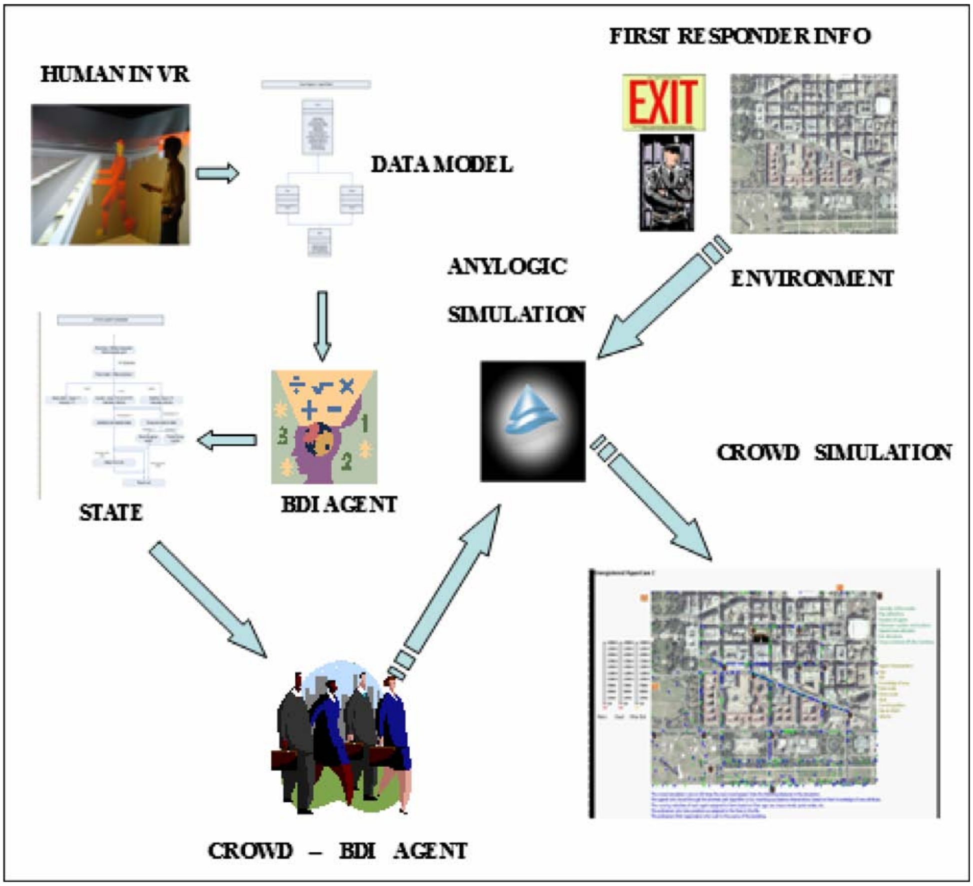

Fig. 2. Overview of crowd simulation development cycle.

4. VR framework

This section explains the different VR methodologies used to conduct VR experiments.

4.1. VR model

In this work, we build the VR model to simulate an emergency scenario in a crowded area of a city for a single intersection with four paths using CAVELib, a library of functions built for the CAVE system. We use OpenGL Performer libraries to create the detailed graphics. Visual C++ is used for programming the graphics in the hardware system. As mentioned before, immersiveness of VR experiments allows us to obtain quasi-real human response data in a very pragmatic way for a potentially life threatening situation without actually putting the human at risk. Note that the human does not actually walk on the floor of the CAVE, but he uses a wand to move around in the environment while standing on the CAVE floor. The program that runs the graphic display of the CAVE system runs separately on each terminal of a Linux cluster. The terminals then are synchronized to run the environment. Some essential modules included in the VR scene include the crowd generated from replicated instances of a 3D model of a human, the smoke/fire which are modeled using multiple continuously moving translucent planes which re-orient themselves with respect to the users view to create the smog effect (the density of which can be varied) and the intersection including policemen and exit signs. The model in VR space along with a sketch of the CAVE system (including walls, projectors and mirrors) is shown in Fig. 3.

4.2. Human behavior (agent characteristics) extraction using VR

The requirements of the crowd simulation considered in this work require building a VR model of a section of the Washington mall area. The VR experimental data are used to determine the trend of human decisions with respect to the choice of exit routes. This collected data is used to populate the BDI agent data model. As mentioned earlier, in order to demonstrate the usefulness of VR in our scenario, we build a prototype VR model consisting of one intersection and outgoing roads. Each

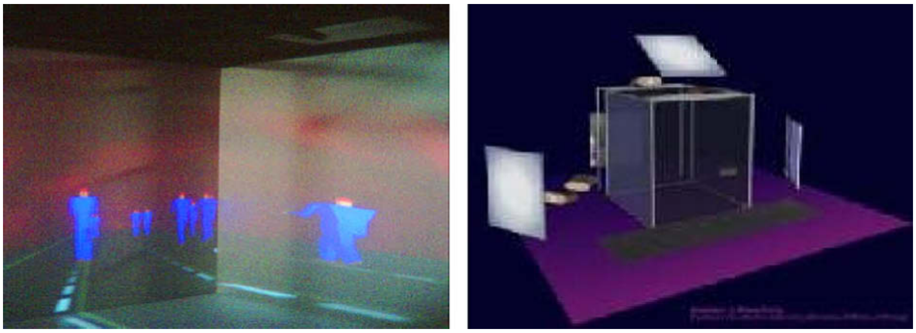

Fig. 3. (a) Snapshot of VR model and (b) CAVE system.

path is then characterized using four parameters: (1) danger (fire) level, (2) crowd, (3) presence of policemen, and (4) presence of exit. The specific aim of the VR experiments is to determine the relative importance given by each human (subject) to each of the above parameters.

The parameters for each path are set to mimic a real emergency scenario in VR. For example, the paths near the explosion have a very high danger level whereas the paths near the exits may have large crowds. The human-in-the-loop experiments reveal the choice of path made by human subjects when faced with imminent danger. It is noted that a confident subject could choose a short yet dangerous path if he thinks he can get through. On the other hand, a prudent subject could choose a path which is less crowded and has less danger even though the exit may not be in sight.

As VR is the next best thing to reality, the perceived values derived from these experiments are the closest approximations that can be obtained without actually emulating a real emergency. The model is verified to ensure that graphics are realistic and no elements exist that distract the subject's immersion. However, note that there exists no method to validate the results obtained as VR experiments are used here for data collection as well. The best way to check the results of the VR experiments is to ensure that the actions of the agent driven by this VR data are acceptable and mimic the results of the VR human-in-the-loop experiments.

Fig. 4 shows the method of quantification of perceptions for each intersection. Based on the number of times a path is taken by the test subjects in VR for each scenario, a weight value is calculated for each parameter to quantize the importance given by humans to that specific parameter. This weight is the relative importance that the subjects perceive for each parameter. Each subject runs through the model 12 times, each time for a different scenario and a total of five subjects are used. This hence gives us 60 decisions points to calculate each weight.

The perception of each path parameter so calculated is used as an agent characteristic in the crowd simulation model to evaluate the probability of picking a path when an agent encounters a given intersection. For example, if an agent in the crowd simulation has to make a choice between four paths, the actual value of each parameter of a path is combined with the agent's perception of each parameter, summed and divided by the total weights on that intersection. The probability of picking each path is the resulting quotient. Thus in the agent-based simulation when an agent arrives at an intersection, it chooses the next path using the probability value computed as above.

5. VR driven BDI agent modeling framework

5.1. Extended BDI architecture

As mentioned before, we propose to use extended BDI framework [15] to model human-like behavior of agents present in the crowd simulation. The framework described in Fig. 5 consists of the following key components which were modeled from VR experimental results. The link between VR and the BDI components are described below:

- *Perceptual processor* observes and tries to interpret the environment from the sensory data. The data collected about perceptions made by the human subjects in VR environment serves as an input to the rule-based algorithm in the perceptual processor module. Based on the value of the perceptions calculated for each of the above parameters, the perceptual processor for every agent computes the beliefs about each path. Various techniques such as Markov chains, Bayesian belief network (BBN) and rule-based systems can be used to carry out this conversion. The choice of the technique for this conversion can be made as per the scenario under consideration. We used a rule-based algorithm in this implementation given the ease of implementation and customization capability. However, work has been initiated to increase the fidelity of the agent model by using Bayesian belief networks implemented through MSBNx (software for BBN). Bayesian belief network (BBN) more closely seem to represent the real human perception process and have hence been preferred over Markov Chains.
- *Cognitive processor* chooses the state of affairs the agent wants to achieve based on the current beliefs and initial intentions. In the considered scenario, the current state (e.g., occurrence of explosion) affects the agent setting its desire to pick

To calculate the perceptions of the agent,

$$Ch_{pi} \Rightarrow Number\ of\ subjects\ who\ pick\ path\ p\ on\ the\ i^{th}\ scenario$$

$$Perc_F \Rightarrow Perception\ of\ fire \Rightarrow \frac{\sum_i \sum_p (Ch_{pi} \times F_{pi})}{\sum_i \sum_p (Ch_{pi} \times (F_{pi} + C_{pi} + P_{pi} + E_{pi}))}$$

$$Perc_C \Rightarrow Perception\ of\ crowd \Rightarrow \frac{\sum_i \sum_p (Ch_{pi} \times C_{pi})}{\sum_i \sum_p (Ch_{pi} \times (F_{pi} + C_{pi} + P_{pi} + E_{pi}))}$$

$$Perc_P \Rightarrow Perception\ of\ policemen \Rightarrow \frac{\sum_i \sum_p (Ch_{pi} \times P_{pi})}{\sum_i \sum_p (Ch_{pi} \times (F_{pi} + C_{pi} + P_{pi} + E_{pi}))}$$

$$Perc_E \Rightarrow Perception\ of\ exits \Rightarrow \frac{\sum_i \sum_p (Ch_{pi} \times E_{pi})}{\sum_i \sum_p (Ch_{pi} \times (F_{pi} + C_{pi} + P_{pi} + E_{pi}))}$$

Now given any path p in a intersection with k paths with characteristics $F_p\ , C_p\ , P_p\ , E_p$

$$probability\ of\ picking\ path\ p \Rightarrow \frac{Perc_F \times F_p + Perc_C \times C_p + Perc_P \times P_p + Perc_E \times E_p}{\sum_{m=1}^{m=k} (Perc_F \times F_m + Perc_C \times C_m + Perc_P \times P_m + Perc_E \times E_m)}$$

$$where, 1 \le p \le k$$

$$F_{pi} \Rightarrow Level\ of\ fire\ on\ a\ path\ p\ for\ the\ i^{th}\ scenario$$

$$C_{pi} \Rightarrow Level\ of\ crowd\ on\ a\ path\ p\ for\ the\ i^{th}\ scenario$$

$$P_{pi} \Rightarrow Level\ of\ policemen\ on\ a\ path\ p\ for\ the\ i^{th}\ scenario$$

$$E_{pi} \Rightarrow Level\ of\ exits\ on\ a\ path\ p\ for\ the\ i^{th}\ scenario$$

Fig. 4. Perception and probability calculations for data from VR experiments.

either a destination goal such as an office or mall (during normalcy) or an exit (during evacuation). The *Deliberator* filters desires to a single intention. In the considered scenario (during evacuation), the deliberator can choose an intention, such as reaching a policeman, one of the exits, the Metro station or any intermediate intersection with the optimism of finding a policeman at that intersection.

- *Real time planner* (RTP) generates different plausible plans to achieve the selected intention based on the current beliefs and intentions. A plan is defined as a sequence of actions. It should also be noticed that more than one plan can be generated for a given case. In the considered scenario, the real time planner uses the knowledge of area (KOA) rating of the agent to provide the shortest path to an exit (KOA = 1), or a number of possible paths to intermediate intersections (KOA < 1).
- *Decision executor* selects one of several plans generated by the real time planner. The policy to select the plan varies from case to case. The decisions taken in response to a given scenario vary from human to human. This is taken into account by using a discrete probability distribution to drive the decisions made by every agent. This implies that the final decision made by an agent for picking a path is unique to that agent (hence making the behavior during the evacuation distinct for each agent). For example, let us suppose for an agent, arriving at an intersection has the following distribution assigned: DISCRETE (0.23:P1, 0.37:P2, 0.3:P3, 0.1:P4). Then the agent arriving at the same intersection 100 times would

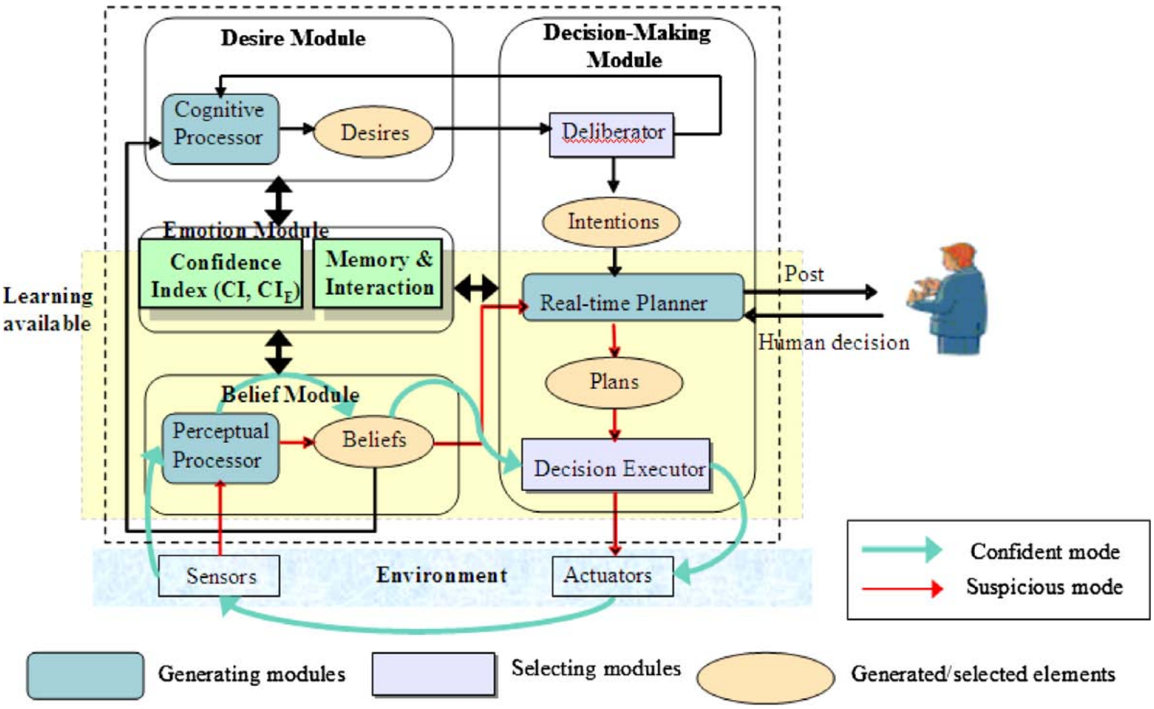

Fig. 5. Extended BDI architecture [15].

pick P1 (path number 1) 23 times even though path 2 may be the best path to take. However, note that the highest probability is assigned to the path that the agent perceives is the best path to take by using the data collected from VR experiments. It is noticed from the experiments that decisions taken by humans are not always the optimal decisions. Hence, this serves as a more realistic approximation of real crowd behavior as mistakes of humans get modeled implicitly.

- *Confidence index* denotes the agent's optimism about achieving its intentions. For example, if the agent performs some major actions or plans successfully, it will be in the *confident* mode. Otherwise, it will be disappointed and will be in the *suspicious* mode. Under the confident mode the agent continues to execute the current plan. The programming for confidence evolution characteristic of each agent was based on the trends observed in VR experiments. Otherwise, in the suspicious mode, the agent tends to be cautious and will reconsider its intentions and reevaluate plans via the real time planner before executing each action.

The simulated human model (BDI agent) continually executes this cycle of observing the environment, deciding which intention(s) to achieve next, determining plan(s) to achieve these intention(s), and then finally executing this plan (a series of actions). In the considered scenario, the confidence index is represented as an aggregation of various attributes such as age, leadership, knowledge of area, velocity, panic state, injury level. A higher confidence index will result in a higher likelihood that an agent will evacuate sooner.

5.2. Development of agents and environment for crowd simulation

This section explains the various techniques used in developing the crowd simulation that conforms to the functional requirements stated earlier. This section also explains the techniques used in realizing the BDI architecture discussed previously (see Section 5.1) and the environment. The key point about the proposed techniques is its generality for use in other simulations of a similar nature.

5.2.1. Modeling behavioral aspects of an agent

The agent's behavior depends essentially on two major factors: the agent's characteristics and the environment characteristics. Every agent in the simulation is given a number of static attributes and a dynamic behavior. Some of the static attributes include age, sex, knowledge of area, panic scale, leadership, independence, injury scale, and current positions. We use empirical discrete distributions to populate these static attributes of agents. For example, 45% of the agents in the model are women and the rest are men. The model was set up to reflect realistic crowd composition. Hence, this research does not consider the sensitivity of the model to these distributions. The attributes such as the goal intersection represent the "intention" part of the BDI framework. The class diagram depicted in Fig. 6(a) formally illustrates the data model for the static attributes

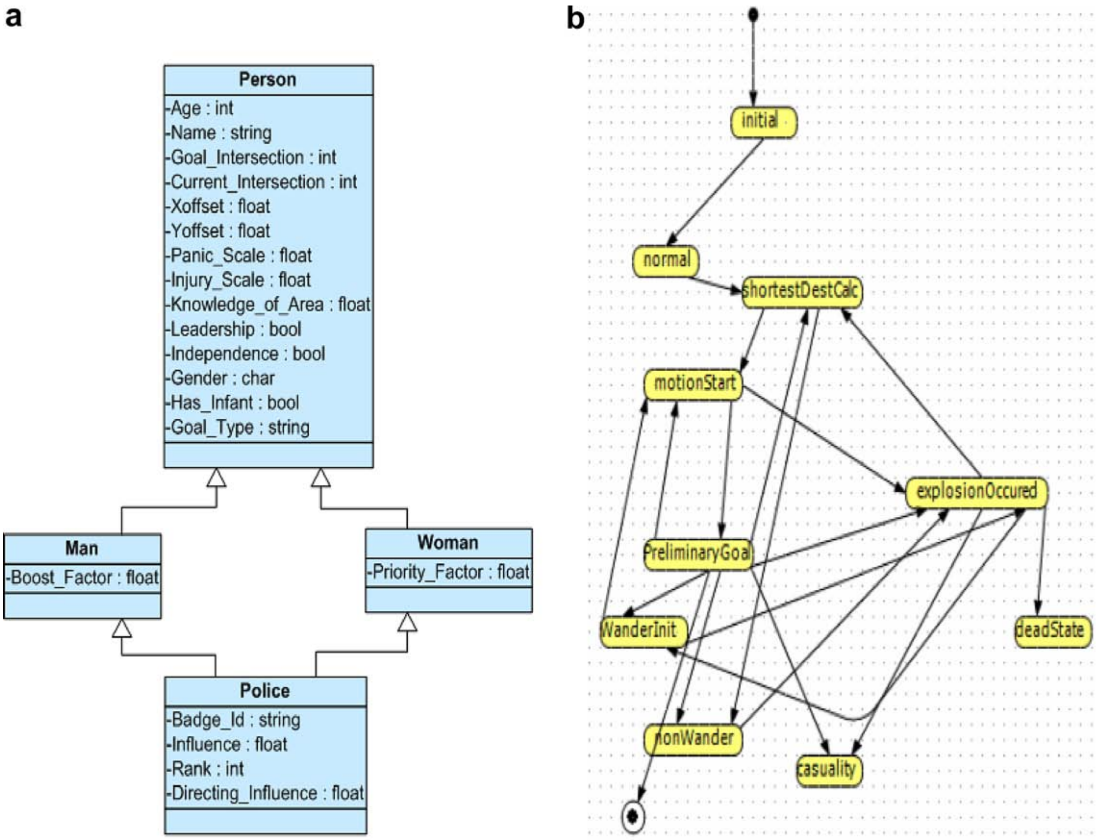

Fig. 6. (a) Class diagram of an agent and (b) state chart diagram of agent.

used in developing the simulation. The class diagram represents an inheritance hierarchy where police is the most specialized type of either men or women.

In this work, the behavior of an agent is modeled using state charts. A State chart diagram is a generic way of representing the behavior of an agent in response to the external events (e.g., explosion) or internal events (e.g., achieving an intention). When an agent reaches the end state, it could be said to have achieved all its intentions. A desire of an agent is modeled in the state chart diagram through a sequence of transitions from one state to another. Depending on the desire, an agent may follow a different sequence of transitions from its current state. Perception of an agent about the current situation is achieved by detecting events occurring in the environment when an agent is in the given state. Thus state chart diagrams can be effectively used to realize the BDI architecture. Fig. 6(b) represents the various states that each agent (person) goes through in our crowd simulation. The arrows in the diagram represent the transition from one state to another (triggered by an internal or external event). Depending on the current state and the belief about the current environment, coupled with its intentions, an agent goes through different transitions to reach different states.

5.2.2. Techniques used for environment modeling

One of the important components of the simulation is the environment in which agents reside or operate. For the considered scenario, the environment of the simulation consists of the National Mall area of Washington DC. In this work, the actual satellite image of the National Mall area is used for simulation. As seen in Fig. 1, the map of Washington DC is divided into a network of roads (horizontal and vertical) and recreational areas (e.g., gardens). It is noted that when people move along the roads in the network, they do not violate the boundary of roads. In Fig. 1, mall areas are present in the lower left and bottom part of the map. In those areas, people can wander arbitrarily without restrictions of road boundaries. After the explosion, the areas near the explosion are cordoned off and people are not allowed to move through this area.

The network of roads is represented as a weighted graph with the vertices being intersections where two roads meet and the distance between two intersections as the weight of the edge. In this work, the adjacency matrix is used to represent the weighted graph (which is the network of roads in the map). Fig. 7 depicts an exemplary representation of a part of a map in terms of an adjacency matrix.

In this work, other areas such as recreational areas are represented as approximated rectangles. The movement of the population in such areas does not follow any well defined path, thus eliminating the need for a specific data structure like

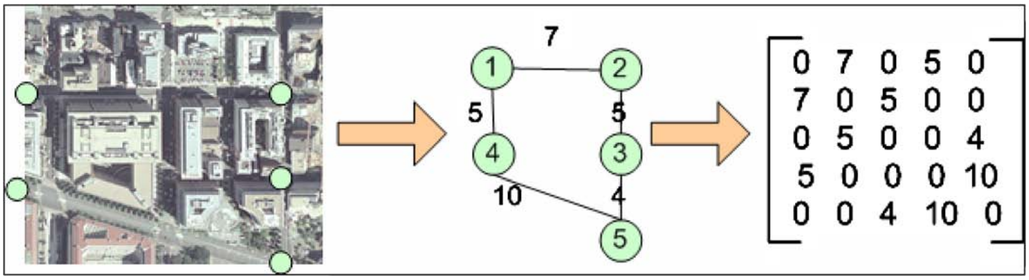

Fig. 7. Adjacency matrix representations from the network of roads in the map.

the adjacency list. The major data structures used to represent most of the input data are maps and vectors. A map data structure simply maps keys to values. While the Vector data structure is used to represent a dynamically extensible array of values. The implementation of these data structures is provided by the built-in java package java.util.

The major focus while designing the environment development process in this section was to keep it so generic that if the map of Washington DC was replaced by another map (area), the changes to the model would be minimal (for example, modifying intersection and path information to reflect the layout of the new map). Hence, all the environment characteristics were represented in a declarative nature using simple text files to describe input that drives the model. More specifically, the following files were used to provide the required input data. Our model makes use of the following declarative files: Map, Intersection Recreational area, Environment, Exits, Police. Some of these files are described below:

- *Map file*: This file describes the considered map as a weighted graph of interconnected network of roads. Each intersection acts as a node in the network.
- *Environment information file*: This file describes the various parameters that characterize specific scenarios (e.g., crowd density in various areas of the map).
- *Police information file*: This file contains information about policemen such as their initial positions and total number of policemen.

It is believed that the above generic data structure to represent an environment will facilitate the development of simulations for similar scenarios. As stated earlier in this paper, the output produced by this simulation may be used by other simulations. Hence, the results of the simulation can be stored into a text file with format agreed upon previously or some well known formats like XML. Use of such files facilitates frictionless communication between different simulations, thus increasing interoperability.

We have tested the generality of the modeling methods proposed in this paper by using different sites (e.g., Phoenix Sky Harbor airport) under attack. In order to adopt our simulation for a different site, the changes that were significant were relevant only to the declarative files mentioned above. After making these changes, only minimal changes were required to change the actual simulation code itself.

5.2.3. Overview of algorithms

The development of the crowd simulation (see Section 5.2.1) involved developing new algorithms and employing the preexisting ones. Among them, two major ones (shortest path algorithm and agent processing algorithm) are explained in brief in this section.

An agent roughly follows the following steps in its lifetime: Read declarative input files, instantiate agents, setup agents with VR extracted data, assign confidence index, start reacting to the explosion, assign panic values, assess the situation, if leader is present follow the leader, move towards the next intersection and stop if end state is reached.

As explained in Section 5.2.2, some people (who are familiar with the area) in the crowd simulation find the shortest path to the exit point which they want to reach. In this work, we employ Dijkstra's [1] shortest path algorithm to calculate the shortest path. We have selected Dijkstra's algorithm because of its simplicity and quadratic computational cost. More specifically, the time complexity of Dijkstra's algorithm used in our case is where represents the number of vertices in the graph. Taking into account various factors mentioned until now, we design the consolidated sequence of various steps that an agent goes through during its lifetime in the simulation.

6. Experiments and results

As mentioned earlier, in this work, the experiments are conducted in two phases. Phase 1 focuses on VR human-in-the-loop experiments (see Section 4.2). Phase 2 involves running the crowd simulations to estimate evacuation key performance indicators (e.g., number of casualties, crowd evacuation time, number of people exiting through the Metro station) of our interest. In general, in order to run the crowd simulation consisting of a crowd of 30,000 people to completion requires approximately 15 min of simulation execution time. The replication length in the exactly same setup is 320 min.

6.1. VR experiments

As mentioned earlier, an intersection model of a section of the map is developed in the VR CAVE. The immersive environment is essential to run human in the loop experiments to extract information about instantaneous human behavior and response to the virtual emergency. The paths leading to and from the intersection are built to mimic the real world emergency situation with varying levels of fire, crowd, policemen, and exit. As mentioned earlier in Section 4.1 each human subject is provided with stereoscopic glasses to create immersion and allowed to navigate through the emergency scenario using a wand.

Experiments have been conducted immersing five subjects in the scenario 12 times each with randomized path graphics. Note that these are intended to be pilot experiments that explore the use of VR for application in crowd simulation, rather than full-scale experiments. Hence, confidence intervals and other statistical measures were not considered for these experiments. This aspect of our research will be the next venture in future research. Each subject was made to run through the environment 12 times to ensure that a one time decision of the human was not accounted for as a character trait. Hence, this provides us with relevant statistical data about each subject's behavior.

During the experiments, each subject was immersed in VR for no more than 1 h. The subjects were all University of Arizona graduate students. Factors such as age were not taken into account using this subject group. However, we plan to include subjects outside the student community comprising of different age groups and backgrounds as a part of extended VR experiments in the future. The particular parameter of interest that is collected is the "path chosen" from which perceptions are calculated. The importance of each parameter (perception fire, crowd, policemen, and exit) obtained from VR is shown in Table 1. A weight of 0.28 for the exit characteristic of the path means that this characteristic influences 28% of the human's perception. Another way to view this table is, if a human encounters a crowded path with an exit, he is more inclined towards taking a path with a lesser crowd rather than take an exit path (sample of an inference drawn from the set of VR experiments conducted).

Table 2 shows an example of probabilities calculated for an intersection with four paths. The numerical value indicates the conduciveness of that characteristic to the path being safe. It is noted that for crowd and fire, a larger number indicates lower intensity. A value of 8/10 for fire means the fire is about to die out. A value of 1/10 means that the path is ablaze. Similarly, path 1 is half full (or half crowded), i.e., the weight of the crowd on the path is 5/10. To demonstrate the importance of VR experiments, probabilities of taking each path are calculated in three different ways: (1) based on VR experiments, (2) calculation based on the weights, and (3) random (uniform) pick. In the first case (VR experiments), we calculate the effect of each characteristic on human perception and use it as a weight on each characteristic weight. For example, path 3 in Table 2 has a probability 21% of getting picked when based on axiomatic (direct – sum of path's characteristics divided by sum of all characteristics of all paths in the intersection) calculations but only 19% when based on calculation from real human data

Table 1
Perception data obtained from VR

Characteristic	Weight
Danger/Fire	0.19
Exit	0.28
Police	0.19
Crowd	0.33

Table 2
Probability calculation for use in the discrete distribution

Paths	Crowd (10)	Fire (10)	Police (10)	Exit (10)	Probabilities obtained using		
					VR	Direct calculation	Random
Path 1	5	5	3	10	0.26	0.26	0.25
Path 2	5	8	3	10	0.29	0.29	0.25
Path 3	5	7	7	0	0.19	0.21	0.25
Path 4	5	7	0	10	0.26	0.24	0.25

Table 3a
Simulation results (population factor)

Population factor (10 policemen, 5 exits)			
Population	10,000	20,000	30,000
Number of casualties	928	1391	1418
Crowd evacuation time	44.81	55.62	74.65
Crowd at Metro	4155	6280	7151

from VR (see Section 4.2 for detailed calculations). The second way directly uses the weights on every characteristic, summing for each path and dividing each obtained value by the total weight on the intersection. It is seen that the probabilities obtained from the real human decisions are different from the other two, and obviously would make the agent mimic the human behavior more closely.

The following observations are made by analyzing the data from the VR experiments:

- The VR experiments were immersive and provided the human with a simulated environment of the real emergency so that their perception could be quantified with a fair degree of realism.
- Most humans tend to make non-optimal decisions. In most cases the second best path was picked. Decisions varied from person to person.
- It was observed that the presence of an exit sign affected the human thought process the most. It was hence the most influential path characteristic. The perception of fire was a close second in influencing human decisions.

Table 3b
Simulation results (policeman factor)

Number of policemen factor (10,000 population, 5 exits)			
Number of policemen	10	15	20
Crowd evacuation time	41.11	36.38	32.62
Crowd at Metro	4180	4912	3145

Table 3c
Simulation results (number of exits factor)

Number of exits factor (10,000 population, 10 policemen)			
Number of exits	5	8	11
Crowd evacuation time	42.12	29.99	26.87
Crowd at Metro	4231	3658	3156

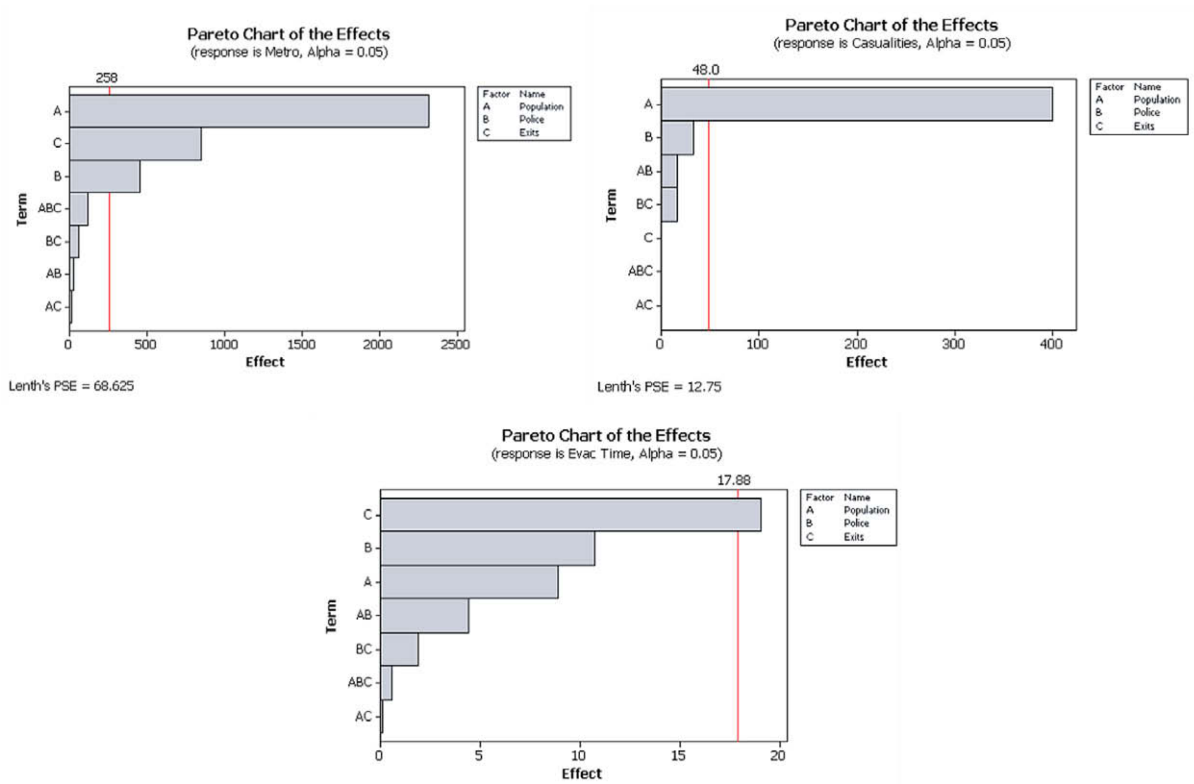

Fig. 8. Pareto charts showing effect of factors on responses.

- The probabilities of picking a particular path drawn from the VR experiment differed considerably from using random probabilities or direct weighted calculations.

6.2. Crowd simulation experiments

Phase 2 involves the experiments with the crowd simulation model run in AnyLogic to obtain information regarding evacuation trends. In this section, we present the results for the effect of various factors on the simulation responses (metrics). We considered a factorial design of experiments (DOE) with three factors and three responses. The factors considered in the data presented above are the population in the area (Table 3a), the number of policemen in the area (Table 3b), the number of exits (Table 3c), the intensity of the explosion, and the distribution of policemen. The simulation metrics considered are (1) number of casualties, (2) crowd evacuation time, and (3) the number of people exiting through the Metro station.

The outcome of the DOE helps us to understand the importance of each factor on each response and is presented below. The DOE was conducted by setting two levels (high and low) of each factor. The Pareto charts for each response is shown in Fig. 8. We observe that the evacuation times and the crowd at Metro depend on all the three considered factors. However, as expected casualties depend only on the population factor. Note that the casualties (deaths) occur only at the start of the simulation, when the explosion happens. Deaths due to injury are not considered and hence casualties are not affected by number of exits or number of policemen; they depend only on the population itself. We also infer that interactions/combinations between each of the factors do not play an important role in the responses. The population factor plays the most important role in determining the magnitude of each of the responses. The number of exits is the second most important factor for evacuation time and crowd at Metro. However, not only is number of exits not important in determining the number of causalities, but it ranks fifth among factor combinations. The number of policemen factor is important given that they help the agents find the shortest path and keep them out of highly dangerous paths along with regulating crowd at Metro. This is ascertained by the fact that it is a crucial factor in determining evacuation times and crowd at Metro.

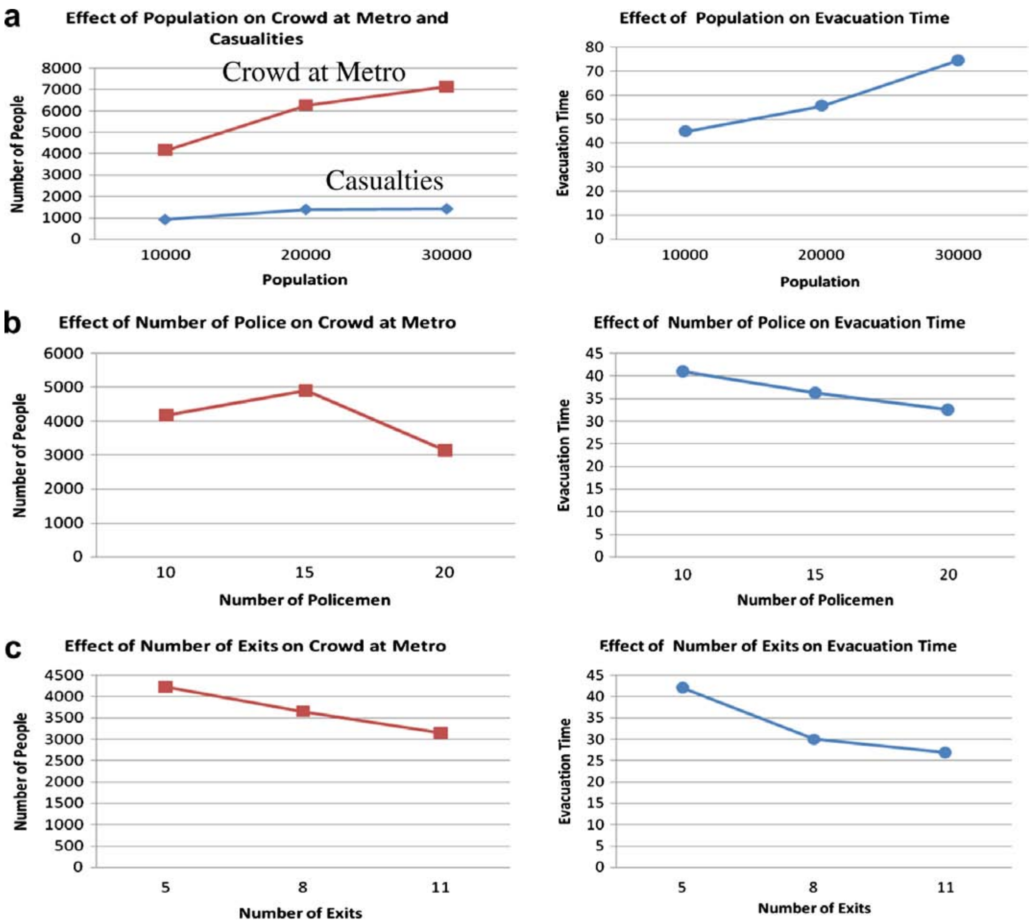

Fig. 9. Graphical representation of Simulation results for the three factors: (a) population, (b) number of policemen and (c) number of exits.

Given the above DOE, experiments were conducted and data was collected to plot graphs showing the trend of responses for various each of the individual factors and to help us make policy recommendations. It was determined that 10 replications were required to achieve a half width of 2 min (considered acceptable) for evacuation times at 95% confidence. All results presented below are averages with 95% confidence.

The base crowd simulation model is set up with 10,000 people, 10 policemen, and 5 exits. Fig. 9(a) depicts graphically, the effect of the population on output metrics like evacuation time and rate at which crowd appears at the Metro. Fig. 9(b) demonstrates the effect of number of policemen on the same output metrics. Fig. 9(c) represents the effect of number of exits on the evacuation time, casualties and crowd at Metro. As expected, number of casualties and crowd at Metro increase with population (see Fig. 9(a)). The evacuation time also increases as it takes more time to evacuate more number of people (see Fig. 9(b) and (c)). Increasing the number of policemen and exits in the system leads to a reduction in evacuation time. However the magnitude of the effect reduces as more and more policemen are introduced (trend is not directly proportional), as there is a minimum evacuation time even with excessive number of policemen. The above trend applies to the number of exits also. The number of people at the Metro seems to be a maximum (local) when an intermediate number of policemen are used. This is attributed to the ratio of number of policemen guiding people to the Metro exit to the number of people approaching each policeman, which is maximum when there is an intermediate number (15) of policemen in the area. Given that there are four outer exits and only one Metro exit (in the base model) and the above results, we conclude that the spread of policemen on the map plays an important role in determining crowd at the Metro. As expected, the average evacuation time for optimal agents were found to be lesser than crowd evacuation time. For the base model, the average evacuation times for optimal agents was 24.33 min. We expect this behavior (by trend) not to change for other factor combinations as well. These results are vital for making policy recommendations in relation to emergency management (see Section 7).

7. Conclusions and future research

In this work, we proposed a novel crowd simulation development methodology, where the BDI agents driven by the human behavior data extracted from the VR experiments are used to build a crowd simulation. The crowd simulation has been applied to study evacuation situation in the event of a terror attack at the National Mall area in Washington DC. Experiments were conducted to explore evacuation scenarios under various environmental conditions.

The results obtained from the VR experiments provided us with an insight into the patterns of human thought processes. We observed that when immersed in a virtual emergency environment, humans tended to perceive paths with exit signs as paths with greatest chance of escape, even if they were slightly more dangerous than paths without exit signs (i.e., even if the path had fire in it as compared with other paths that have less or no fire). Danger (fire), police and crowd also influenced the perception of the humans in the decreasing order after the exit signs.

The crowd simulation constructed using the proposed modeling scheme allowed us to examine the effect of various factors (the number of people in the area (population), the number of policeman in the area, the number of exits, the intensity of the explosion, and the distribution of policemen) on the simulation metrics (number of casualties, crowd evacuation time, and the number of people exiting through the Metro station). From the experiment, the best exit routes and congestion areas have been identified. During evacuation, the intersections to the bottom left of the map shown in Fig. 1 were found to be most congested due to the presence of more exits and proximity to the explosion site. We recommend deployment of policemen in congested areas and popular routes to aid evacuation. Further research work can evaluate the quantitative impact of placement of policemen in vantage positions based on overall goal (whether to minimize or maximize crowd at Metro). We also believe that the presence of emergency transportation at key areas close to the evacuation will reduce the time of evacuation, since the transportation can act like guides that direct people to evacuate faster. We recommend having more exit points where feasible, in other words, a fair distribution of crowd as a part of guidance policy for first responders. This will facilitate a better crowd management at Metro stations and other key exits and for setting Metro schedules in such emergency scenarios.

Further research must be undertaken involving transportation of injured people to the hospitals, where the role of policemen is expected to affect the number of casualties. The number of policemen in the area was found not to affect the number of people exiting through the Metro, but was found to cause a sharp drop in evacuation time. The versatility of the model makes it suitable for extension to other cities, scenarios and situations. It is believed that the extended BDI agent considered in this work can be used to mimic human behavior in a variety of other applications such as agents for driverless cars and predicting sporting outcomes.

Future work will also involve addressing computational issues, involving more accurate models and other constraints of the model. The model can be improved to better predict the number of casualties by considering factors such as death due to chaos, and aggravation of sustained injury. Also we plan to consider other factors (different number of group leaders, different emergency responders such as ambulances, etc.) involved in the crowd simulation and also scenarios of multiple bombings (where the goals are to minimize the probability of being injured by the second bombing as well as to minimize the evacuation time). Policies about detailed first responder configurations and resource utilizations can be made by extending the present model. A sensitivity analysis to analyze effect of different population compositions can be set up. A major constraint of the VR experiments is the unavailability of real data to validate the results. Besides, the VR method is constrained by time of exposure also that restricts us from conducting many replications. Also there is limited opportunity (besides VR) to validate the agent model against complex cognitive human behavior and special cases seen during emergencies. The data drawn from the VR model can be enhanced by incorporating different types of agents by classifying results obtained from VR experiments by behavior into groups of types of people (aggressive, calm, etc.). However, we would need a much larger subject population for the same, and we have already undertaken work on this. The realization of perceptual processor can be evaluated by using better probabilistic models, such as Bayesian Belief Networks. The empirical evidence obtained from these experiments can result in more accurate modeling of human behavior. As for the VR experiments, we intend to further increase the scope of application of VR by involving more sophisticated feature extraction and involving more subjects.

Acknowledgement

This work was supported by Air Force Office of Scientific Research under AFOSR/MURI F49620-03-1-0377 and National Institute of Standards and Technology Grant under SB1341-05-W-0852.

References

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., MIT Press/McGraw-Hill, Cambridge/New York, 2001. pp. 580–642 (Chapters 24 and 25).
[2] K.H. Dam, M. Winikoff, Comparing agent-oriented methodologies, in: Proceedings of AOIS 2003, Lecture Notes in Computer Science, vol. 3030, Springer, 2004, pp. 78–93.
[3] R.A. Earnshaw, J.A. Vince, H. Jones, Virtual Reality Applications, Academic Press Limited, New York, 1995.
[4] R. Fikes, N. Nilsson, STRIPS: a new approach to the application of theorem proving, Artificial Intelligence 2 (1971) 189–208.
[5] T. Hamagami, H. Hirata, Method of crowd simulation by using multiagent on cellular automata, in: Proceedings of IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03), 2003.
[6] S. Jain, C.R. McLean, An integrated framework for modeling and simulation for incident management, Journal of Homeland Security and Emergency Management 3 (1) (2006).
[7] D. Kinny, M. Georgeff, A. Rao, A methodology and modeling technique for systems of BDI agents, in: Proceedings of the Seventh European Workshop on Modeling Autonomous Agents in a Multi-Agent World, MAAMAW'96, Lecture Notes in Artificial Intelligence, Springer, New York, 1996, p. 1038.
[8] R. Laughery, Computer simulations as a tool for studying human-centered systems, in: Proceedings of the 2001 Winter Simulation Conference, 1998, pp. 61–65.
[9] D. McGrath, A. Hunt, M. Bates, A simple distributed simulation architecture for emergency exercises, in: Ninth IEEE International Symposium on Distributed Simulation and Real Time Applications, 2005.
[10] E. Norling, Folk psychology for human modeling: extending the BDI paradigm, in: International Conference on Autonomous Agents and Multi Agent System (AAMAS), New York, 2004.
[11] E. Norling, Capturing the quake player: using a BDI agent to model human behavior, in: Proceedings of the Second Isnternational Joint Conference on Autonomous Agents and multiagent systems, 2003, pp. 14-18.
[12] K. Pimentel, K. Teixera, Virtual Reality: Through the New Looking Glass, Intel/McGraw-Hill, New York, 1994.
[13] A.S. Rao, M.P. Georgeff, BDI agents: from theory to practice, Technical Report 56, Australian artificial intelligence institute, Melbourne, Australia, 1995.
[14] A.S. Rao, M.P. Georgeff, Decision procedures for BDI logics, Journal of Logic and Computation 8 (1998) 293–342.
[15] X. Zhao, Y. Son, BDI based human decision making model in automated manufacturing system, International Journal of Modeling and Simulation, in press.