An Integrated Human Decision Making Model for Evacuation Scenarios under a BDI Framework

SEUNGHO LEE and YOUNG-JUN SON
The University of Arizona
and
JUDY JIN
University of Michigan

An integrated Belief-Desire-Intention (BDI) modeling framework is proposed for human decision making and planning for evacuation scenarios, whose submodules are based on a Bayesian Belief Network (BBN), Decision-Field-Theory (DFT), and a Probabilistic Depth-First Search (PDFS) technique. A key novelty of the proposed model is its ability to represent both the human decision-making and decision-planning functions in a unified framework. To mimic realistic human behaviors, attributes of the BDI framework are reverse-engineered from human-in-the-loop experiments conducted in the Cave Automatic Virtual Environment (CAVE). The proposed modeling framework is demonstrated for a human's evacuation behaviors in response to a terrorist bomb attack. The simulated environment and agents (models of humans) conforming to the proposed BDI framework are implemented in AnyLogic®agent-based simulation software, where each agent calls external Netica BBN software to perform its perceptual processing function and Soar software to perform its real-time planning and decision-execution functions. The constructed simulation has been used to test the impact of several factors (e.g., demographics, number of police officers, information sharing via speakers) on evacuation performance (e.g., average evacuation time, percentage of casualties).

Categories and Subject Descriptors: K 4.1 [**Computers and Society**]: Public Policy Issues—*Human safety*; H 1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing*

General Terms: Algorithms, Human Factors, Experimentation, Management

Additional Key Words and Phrases: Emergency evacuation, BDI, human decision behavior, planning, Bayesian belief network

This work was supported by the Air Force Office of Scientific Research under AFOSR/MURI F49620-03-1-0377 and the Arizona Chapter of the Human Factors and Ergonomics Society.

Authors' addresses: S. Lee and Y.-J. Son, Systems and Industrial Engineering, The University of Arizona, 1127 E. James E. Rogers Way, Engineering Building #20, Tucson, AZ 85721; email: {mountlee,son}@arizona.edu; J. Jin, Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI; email: jhjin@umich.edu.

DOI 10.1145/1842722.1842728 http://doi.acm.org/10.1145/1842722.1842728

ACM Reference Format:
Lee, S., Son, Y.-J, and Jin, J. 2010. An integrated human decision making model for evacuation scenarios under a BDI framework. ACM Trans. Model. Comput. Simul. 20, 4, Article 23 (October 2010), 24 pages. DOI = 10.1145/1842722.1842728 http://doi.acm.org/10.1145/1842722.1842728

1. INTRODUCTION

Human decision behaviors have been studied by researchers in various disciplines such as artificial intelligence, psychology, cognitive science, and decision science [Lee et al. 2008]. As a result of those efforts, several models have been developed to mimic human decision behaviors. Lee et al. [2008] classified these models into three major categories based upon their theoretical approach: (1) an economics-based approach, (2) a psychology-based approach, and (3) a synthetic engineering-based approach. Each approach exhibits strengths and limitations. First, models employing the economics-based approach have a concrete foundation, based largely on the assumption that decision makers are rational [Mosteller and Nogee 1951; Simon 1955; Opaluch and Segerson 1989; Gibson et al. 1997]. However, one limitation is their inability to represent the nature of human cognition (e.g., stress, fatigue, and memory). To overcome this limitation, models using a psychology-based approach (second category) have been proposed [Edwards 1954; Einhorn 1970; Payne 1982; Busemeyer and Diederich 2002]. While these models explicitly account for human cognition, they generally address human behaviors only under simplified and controlled laboratory conditions. As people are seldom confined to the conditions of static laboratory decision problems, those models may not be directly applicable to human behaviors in a more complex environment [Rothrock and Yin 2008]. Finally, the synthetic engineering-based models, which complement economics- and psychology-based models, employ a number of engineering methodologies and technologies to help reverse-engineer and represent human behaviors in complex and realistic environments [Laird et al. 1987; Newell 1990; Rao and Goergeff 1998; Konar and Chakraborty 2005; Sirbiladze and Gachechiladze 2005; Zhao and Son 2007; Rothrock and Yin 2008; Lee et al. 2008]. The human decision-making models in this category consist of engineering techniques used to implement submodules. However, given all of the possible interactions between submodules, the complexity of such comprehensive models makes them difficult to validate against real human decisions. More recently, a growing number of interdisciplinary studies have been conducted to complement each of the preceding categories [Shizgal 1997; Sanfey et al. 2006; Glimcher 2003; Sen et al. 2008; Gao and Lee 2006]. In this article, we propose a novel, comprehensive model of human decision-making behavior, effectively integrating engineering-, psychology-, and economics-based models. Another novelty of the proposed model is its ability to represent both the human decision-making and decision-planning functions in a unified framework.

Soar, Act-R, and Belief-Desire-Intention (BDI) are three popular synthetic engineering-based models from which we could develop a more comprehensive, modular, and computational human decision model. Soar and Act-R have their theoretical bases in the unified theories of cognition [Newell 1990], an

effort to integrate research from various disciplines to describe a single human cognition. Thus, Soar and Act-R concentrate on the actual mechanisms of the brain during information processing, including tasks such as reasoning, planning, problem solving, and learning. Consequently, these models become complex and difficult to understand. On the other hand, the core concepts of the BDI paradigm, originally based in folk psychology, allow use of a programming language to describe human reasoning and actions in everyday life [Norling 2004]. Because of this straightforward representation, the BDI paradigm can easily map extracted human knowledge into its framework. Accordingly, the BDI paradigm has been applied successfully in many medium-to-large-scale software systems including an air-traffic management system [Kinny et al. 1996]. For this reason, we have adopted the BDI framework as a core modeling and integration tool in our research.

In this article, the proposed human decision model is illustrated using scenarios of emergency evacuation from a terrorist bombing attack in a large city. Effective crowd management requires accurate prediction of the impact of such incidents on the crowd as well as on the environment (which will affect the crowd's behavior). Furthermore, the human lives at stake require that such predictions be highly accurate. For these purposes, high-fidelity simulation is an ideal technique, as it enables experiments not feasible during real incidents. In this article, we construct a model of an individual with unique characteristics (i.e., situation awareness) based on information extracted from human-in-the-loop experiments. Those characteristics are instantiated as entities with different attribute values to create a crowd, which will act in accordance with the proposed, highly detailed human decision model.

The rest of this article is organized as follows. In Section 2, we discuss the BDI framework and the techniques used in its submodules. In Section 3, the proposed planning algorithm is discussed in detail. Then, Section 4 discusses human-in-the-loop experiments in the Virtual Reality (VR) environment and construction of a crowd simulation that conforms to the proposed human behavior model and to the human behaviors extracted from the experiments. Finally, Section 5 presents conclusions.

2. EXTENDED BELIEF-DESIRE-INTENTION MODEL AND ENABLING TECHNIQUES

BDI is a model of the human reasoning process, where a person's mental state is characterized by three major components: beliefs, desires, and intentions [Rao and Georgeff 1998]. Beliefs are information that a human possesses about a situation, and beliefs may be incomplete or incorrect due to the nature of human perception. Desires are the states of affairs that a human would wish to see manifested. Intentions are desires that a human is committed to achieve. Zhao and Son [2007] extended the decision-making module (corresponding to the intention component) of the original BDI model to include three detailed submodules: (1) a *deliberator*, (2) a *real-time planner*, and (3) a *decision executor* in the decision-making (intention) module (see Figure 1). This extension was necessary to accommodate both the decision-making and decision-planning

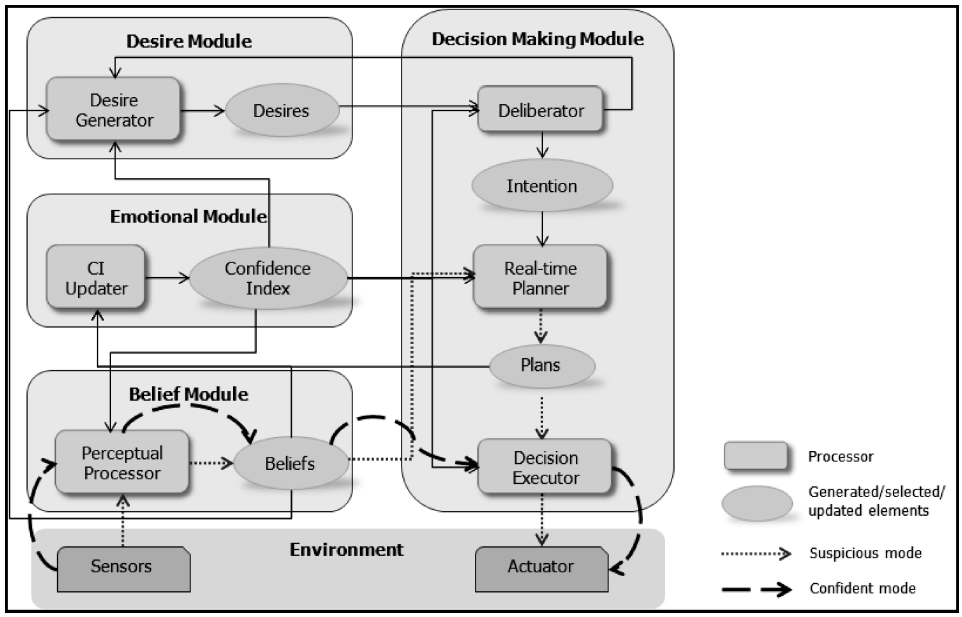


Fig. 1. Components of the extended BDI framework.

functions in a unified framework. In addition, an emotional module containing a *confidence index* and *time pressure* also has been appended to represent these aspects of human psychology. The emotional module affects and is affected by the three other mental modules, that is, beliefs, desires, and decision making. While Zhao and Son's [2007] achievement was to provide a conceptual extension of the BDI model, this article discusses actual algorithms and techniques that we have employed and further developed to realize submodules for the extended model. While decision-making behavior in the extended BDI model is an ongoing and iterative process, we can start from the belief module for purposes of illustration (see Figure 1). The *perceptual processor*, a subjective information filter in the belief module, translates information about the environment and the agent (human model) itself into its beliefs. As a result, the agent has only partial and possibly biased information about the environment and itself. This information therefore is labeled *belief*, not knowledge. Then, based on the current *beliefs*, the agent evaluates potential states of affairs and finds desirable states (*desires*) through the *desire generator*. The agent selects one *desire* (which becomes an *intention*) via the *deliberator*. The agent then generates alternative *plans* based on its current *beliefs* with the purpose of achieving its *intention*. A *plan* in this article is defined as a sequence of actions the agent needs to perform to achieve its *intention*. Once an optimal or satisfactory *plan* (a series of tasks/actions) is identified, a *decision executor* in the decision-making module executes the series of tasks specified in the *plan*. In this work, a *confidence index* is an exponential smoothing function of the deviation between what is predicted about the environment during the planning stage and the actual environment during the execution stage. If the *confidence*

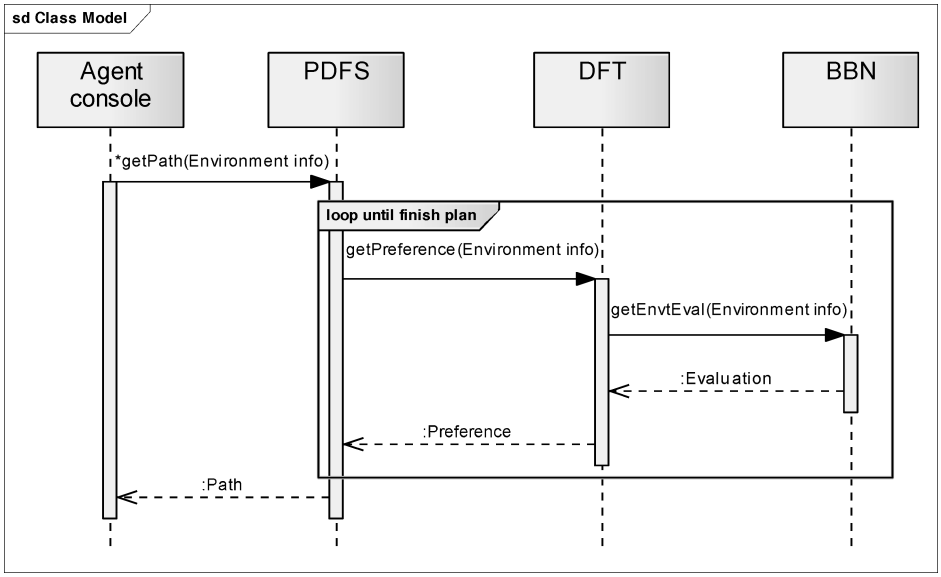


Fig. 2. Sequence diagram of components (corresponding techniques) of the human behavior model.

index exceeds a certain threshold (so that the model is operating in the so-called "confident" mode), then the *decision executor* executes all the tasks in the plan. Otherwise the model is operating in the so-called "suspicious" mode, and replanning is performed before executing each task.

In this article, we employed and further developed novel techniques from various disciplines to realize and implement each component of the extended BDI modeling framework, including the Bayesian Belief Network (BBN), Decision Field Theory (DFT), and Probabilistic Depth-First Search (PDFS). Each of these techniques is explained in the following subsections. Figure 2 depicts a sequence diagram of the overall decision planning process, displaying the sequential interactions between components (and corresponding techniques) of the extended BDI modeling framework. Whenever an agent needs to make a decision, it performs planning (single horizon or multi-horizon) via PDFS, which in turn accesses DFT and the BBN to obtain preferences and assess the environment, respectively. Once DFT obtains an assessment of the environment from the BBN, it calculates the preference value of each option, which will be used to calculate the choice probability of each option. Then PDFS selects an option and makes a plan based on the calculated choice probability. Since the decision has been made based on the preference value of each option with the predicted human preference value provided by DFT, which has been successfully applied to many cognitive tasks [Busemeyer and Diederich 2002], it can mimic the cognitive nature of human decision behavior. However, for applications in which a decision depends solely on rational reasoning, DFT can be replaced with other techniques such as rule-based decision making.

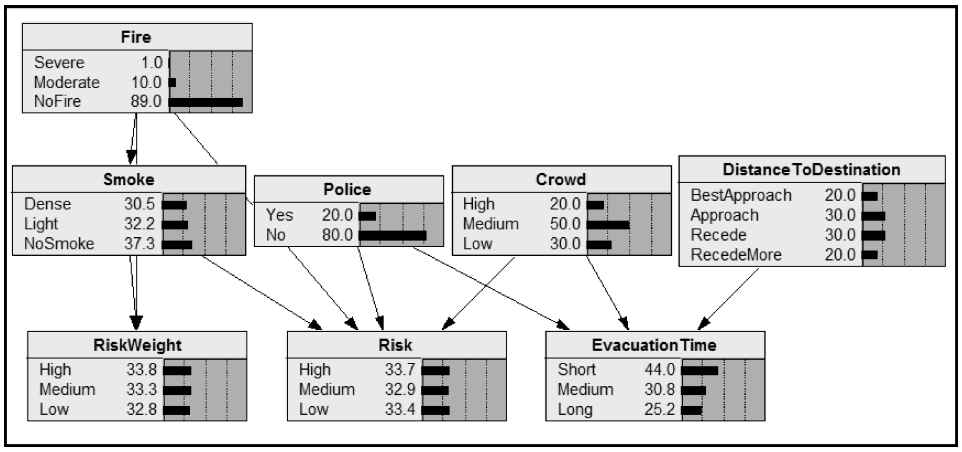


Fig. 3. The BBN for inferring risk, evacuation time, and risk weight for an option.

2.1 Application Scenario

The proposed human decision behavior model is illustrated in the context of crowd evacuation behaviors in response to a terrorist bomb attack in the Washington, D.C., National Mall area. A map (satellite image) of the area is shown in Figure 8 in Section 4.1. Given the scenario, we characterized different types of agents (models of humans) based on: (1) familiarity with the area (which will entail different evacuation planning), (2) risk-taking behavior, (3) confidence index (affecting the moving speed of an agent and leader/follower behavior), and (4) guidance by police. The scenario begins when an explosion occurs, the police are informed of it via radio transmission, and the police ask people (agents) around them to evacuate the area. Agents' evacuation behaviors will vary according to their characteristics. For example, those who are familiar with the area (commuters) invoke the multi-horizon planning algorithm (see Figure 4) to develop their evacuation plan. On the other hand, those who are not familiar with the area (novice agents) move from intersection to intersection (i.e., invoke the single-horizon planning algorithm) and may be guided by a police officer or a commuter agent to the nearest exit. One example of an exit point is a Metro station located well beyond the radius of the explosion (see Figure 8). The *confidence index* of an agent also determines the leader/follower behavior (see Section 4.1). Once agents reach an exit point, they are discarded from our simulation.

2.2 Decision Field Theory (DFT)

Decision Field Theory (DFT) is a human decision-making model based on the principles of psychology rather than economics [Busemeyer and Diederich 2002]. It provides a mathematical framework to represent the psychological preferences of humans with respect to different options during their deliberation process [Busemeyer and Townsend 1993]. Later, Lee et al. [2008] extended DFT to cope with dynamically changing environments by employing BBN, and their model was validated using human-in-the-loop experiments involving a

```
1:  CALL BBN and DFT to get the preferences of PATHs from the current position
2:  IF Agent has knowledge of local paths THEN
3:    REPEAT
4:      SELECT a PATH that is directly connected to the current position based on the
            probability distributed according to preference
5:      SET the preference value of the selected PATH to 'worst'
6:      ADD PATH to list of PATHs
7:      IF the selected PATH forms a cycle, THEN DELETE the PATHs in the cycle
8:      UPDATE current position to the destination end of the PATH
9:      GET the preference for all PATHs that are connected to the current position based on
            current knowledge
10:   UNTIL Agent reaches the destination or has selected a series of n PATHs (where n =
            number of edges)
11: ELSE
12:   SELECT a PATH that is directly connected to the current position based on the probability
            distributed according to preference
13:   SET the preference value of the selected PATH to 'worst'
14: ENDIF
15: RETURN the list of PATH
```

Fig. 4. Planning algorithm.

simulated stock market. In this work, we employ DFT (in conjunction with PDFS as described in Section 2.4) to implement the *real-time planner* submodule in the decision-making module of the proposed extended BDI framework (see Figure 1). In extended DFT, the human preference can be described as Eq. (1).

$$P(t+h) = SP(t) + CM(t+h)W(t+h) \tag{1}$$

In Eq. (1), $P(t)$ (an $m \times 1$ vector, where m is the number of options) represents a preference state, where $P_i(t)$ represents a preference value for option i at time $t < T_D$, where T_D is the time when a final decision is made, and h is a time step. The stability matrix S of Eq. (1) represents the lingering effect of the preference from the previous state (the memory effect) and the effects of interactions among options. In particular, the diagonal elements of S denote the memory of the previous state's preferences, and the off-diagonal elements denote inhibitory interactions among the competing options. The matrix S is assumed to be symmetric with the same value for each diagonal element to represent the fact that each option is subject to the same memory effect and the same interaction effect with each other option. Furthermore, for the stability of this linear system, the eigenvalues λ_i of S are assumed to be less than one in magnitude ($|\lambda_i| < 1$). The value matrix $M(t)$ (an $m \times n$ matrix, where m is the number of options, and n is the number of attributes) represents the subjective evaluations (perceptions) of a decision-maker regarding each attribute of each

option at time t. In other words, given items of objective information (e.g., smoke, fire, police, crowd, and distance in the considered scenario), evacuators obtain their own subjective evaluations for each attribute (e.g., risk and time) of each given option (e.g., a path from an intersection); and these evaluations constitute the $M(t)$ matrix. The weight vector $W(t)$ (an $n\times 1$ vector, where n is the number of attributes) allocates the weights of attention corresponding to each attribute considered at time t. An important assumption of DFT is that the weight vector $W(t)$ changes over time according to a stationary stochastic process. The matrices $M(t)$ and $W(t)$ can be inferred from BBN in our research. The matrix C is the contrast matrix comparing the weighted evaluations of each option, $M(t)W(t)$. If each option were evaluated independently, then C would be I (the $m \times m$ identity matrix). In this case, the preference value of each option could increase simultaneously (see Eq. (1)). Alternately, the elements of the matrix C can be defined as $c_{ii} = 1$ and $c_{ij} = -1/(m-1)$ for $i \neq j$ where m is the number of options. In this case, increasing the preference for one option decreases the preference for all alternative options, and the sum of the elements of $CM(t)W(t)$ (an $m\times 1$ vector, where m is the number of options) is always zero. In this research, we adopted the latter definition of the C matrix. For example, if our evacuation scenario has two options (see Section 2.1), the corresponding DFT formula, as defined in Eq. (1), is shown in Eq. (2).

$$\begin{pmatrix} p_1(t+h) \\ p_2(t+h) \end{pmatrix} = \begin{pmatrix} s_1 & s_2 \\ s_2 & s_1 \end{pmatrix} \begin{pmatrix} p_1(t) \\ p_2(t) \end{pmatrix} + \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \times \begin{pmatrix} m^1_{\text{risk}}(t+h) & m^1_{\text{time}}(t+h) \\ m^2_{\text{risk}}(t+h) & m^2_{\text{time}}(t+h) \end{pmatrix} \begin{pmatrix} w_{\text{risk}}(t+h) \\ w_{\text{time}}(t+h) \end{pmatrix} \tag{2}$$

2.3 Bayesian Belief Network For Perceptual Processor

In this research, a Bayesian Belief Network (BBN) is employed to represent the *perceptual processor* (see Figure 1) of the proposed human behavior model in a dynamically changing environment. The BBN is a cause-and-effect directed acyclic network whose nodes represent the variables to be considered, and whose arc directions encode the conditional dependencies and cause-effect relationship between variables. By using the BBN, we can capture probabilistic relationships as well as historical information between variables by incorporating prior and conditional probabilities that can be used to infer the posterior probabilities using Bayes' theorem. The major advantage of the BBN as a *perceptual processor* is its ability and flexibility to handle uncertain and dynamic environments. In other words, even if not all the information considered by the BBN is currently available, the BBN still can give a convincing probabilistic answer based on historical data.

Figure 3 depicts a BBN used to infer the beliefs of an agent in response to the evacuation scenario described in Section 2.1. The beliefs that the BBN infers from environmental information (e.g., smoke, fire, police, crowd, and distance) consist of: (1) the perceived values of attributes (risk and evacuation time) for the option under consideration (i.e., a path from an intersection), and (2) weights accorded to each attribute (risk versus evacuation time). The weights

associated with each attribute at time t,

$$W(t) = \begin{pmatrix} w_{\text{risk}}(t) \\ w_{\text{time}}(t) \end{pmatrix}$$

can be obtained from the "RiskWeight" node of the BBN in Figure 3 by defining $w_{\text{risk}}(t)$ = "RiskWeight" and $w_{\text{time}}(t) = 1-$ "RiskWeight". Similarly, the perceived value of each attribute of each available option at time t,

$$M(t) = \begin{pmatrix} m^1_{\text{risk}}(t) & m^1_{\text{time}}(t) \\ m^2_{\text{risk}}(t) & m^2_{\text{time}}(t) \\ \vdots & \vdots \end{pmatrix}$$

can be obtained from "Risk" and "EvacuationTime" nodes of BBN by assigning $m^{\text{i}}_{\text{risk}}(t)$ = 'Risk' and $m^{\text{i}}_{\text{time}}(t)$ = "EvacuationTime" for option i. Thus these inferred values for $W(t)$ and $M(t)$ will be fed into Eq. (2) to derive the choice probability via DFT. Although only two attributes (risk and time) are considered in this article, it is noted that additional attributes (e.g., fear of dying, potential for escape, desire to help others) can be incorporated easily. The inferred belief from the BBN is intended to be similar to that of a real human. This similarity can be obtained by constructing the BBN based on the data obtained from human-in-the-loop experiments. Details of the experiment are described in Section 4.

2.4 Probabilistic Depth-First Search (PDFS) for Real-Time Planner

In this research, we employed the Probabilistic Depth-First Search (PDFS) method (in conjunction with DFT) to implement the *real-time planner* in the extended BDI framework (see Figure 1). The depth-first search method requires less information to be tracked compared with the breadth-first search, so is believed to be more appropriate for ordinary people. To this end, we employed Soar software to implement the PDFS. While Soar is also known as a general cognitive architecture, the aspect of Soar employed in this research is the computer programming tool, which provides built-in data structures and operators for depth-first search. Soar searches the problem space in a depth-first manner, where a particular branch is selected based on the choice probability of each branch. In order to evaluate a series of decisions in a plan, Soar first proposes all available options and then selects one of them as the next task based on their preference values. Soar has eleven types of preferences including *acceptable, require, prohibit, reject, better, worse, best, worst, unary indifferent, binary indifferent*, and *numeric indifferent*. Except for the *numeric indifferent* preference, all of the preference types are deterministic. For example, an option with the *require* preference must be selected, and an option with the *prohibit* preference must not be selected. The preferences *better* and *worse* enable the comparison of two options, and the superior option is selected. When the preference is of the *numeric indifferent* type, the numeric values of each option are interpreted as choice probabilities.

As discussed in Section 2.2, DFT provides a method of calculating preference values for each option based on the current environment. Preferences evolve

over a series of time steps as the agent's attention shifts between attributes. At the end of this process (known as DFT evolution), the agent selects the option with the highest preference value. For each decision, we performed 1000 replications of DFT evolution. We calculated the choice probability for each option by counting the proportion of the 1000 replications in which that option ended up with the highest preference value. For the binary choice problem, Lee et al. [2008] proved that there is a finite time t (a duration of DFT evolution) when the choice probability converges. To obtain the converged choice probability, we must perform DFT evolution for a sufficient number of time periods. If we fail to evolve the preferences for enough time periods, the choice probability that we calculate will be inconsistent across DFT deployments. Once we obtain a converged choice probability for each option, we feed it into Soar as the value of a *numeric indifferent* preference. For example, based on the environmental information (I_{smoke}, I_{fire}, I_{police}, I_{crowd}, and I_{distance} in our scenario of Section 2.1) and the available options (O_{right}, O_{left}, O_{forward}, O_{backward} which denote going right, left, forward, and backward, respectively, in our scenario of Section 2.1), DFT evolves the preference values for each option until they converge in each of the 1,000 replications. Let us suppose the numbers of occurrences of each option having the highest preference value are given (p_{right}, p_{left}, p_{forward}, p_{backward}) based on the converged preferences from multiple replications ($p = \sum_{i \in \text{Options}} p_i$). And suppose further that we select an option with the highest preference value as a final decision in each replication. Then, we can calculate the choice probability of each option ($\Pr(O_{\text{right}}) = p_{\text{right}}/p$, $\Pr(O_{\text{left}}) = p_{\text{left}}/p$, $\Pr(O_{\text{forward}}) = p_{\text{forward}}/p$, $\Pr(O_{\text{backward}}) = p_{\text{backward}}/p$) and feed them into Soar as the value of the *numeric indifferent* preference. Then, Soar selects an option based on the given preference (choice probability in this case). The preceding procedure is repeated for the given length of the planning horizon, which differs for each individual human (agent). For example, the planning horizon for a novice agent (who has no knowledge about the area) would be one step (i.e., next path). However, a commuter who is familiar with the environment may have a multi-horizon plan. In this work, the planning horizon of a commuter agent is set randomly to a value between 2 and 10. More details about the multi-horizon planning algorithm are discussed in Section 3.

2.5 Confidence Index

As mentioned before, the *confidence index* determines: (1) the execution mode (confident or suspicious mode) in the BDI framework, and (2) the type of agent (leader or follower). Eq. (3) depicts the confidence index proposed in this research as

$$CI_t = \alpha \cdot e^{-\mathrm{d}_t} + (1 - \alpha) CI_{t-1} \quad \text{for } t = 1, 2, \ldots, \tag{3}$$

where $d_t > 0, 0 \leq \alpha \leq 1$, and $0 \leq CI_0 = \beta \leq 1$. In Eq. (3), d_t denotes the deviation between what is predicted about the environment during the planning stage and the actual environment during the execution stage. Thus, the agent updates its *confidence index* at each street intersection, where it can compare its prediction with an actual observation. Eq. (4) depicts d_t used in our research,

where $m^i_{\text{risk}}(t)$ is the evaluation of the risk associated with option i, and $m^i_{\text{time}}(t)$ is the evaluation of the evacuation time associated with option i.

$$d_t = \left|m^i_{\text{risk}}(t) - m^i_{\text{risk}}(t-1)\right| + \left|m^i_{\text{time}}(t) - m^i_{\text{time}}(t-1)\right| \quad (4)$$

The parameter α in Eq. (3) represents the effect of the agent's previous level of confidence on its current level of confidence, which varies depending on the individual human. The initial confidence value (β) is given and depends upon the agent's type. By definition, the range of the *confidence index* is between 0 and 1. In this work, the initial *confidence index* assigned to commuter agents and novice agents are unif(0.5, 1) and unif(0.2, 0.7), respectively.

3. DETAILED REAL-TIME PLANNING ALGORITHM

This section discusses in greater detail the planning algorithm (both multi-horizon and single-horizon) (see Sections 3.1 and 3.2) implemented in Soar. As mentioned earlier, Soar is a computer programming tool that can be used to implement theories and concepts in various fields of cognitive science such as psychology, linguistics, anthropology, and artificial intelligence. In order to make an architecture (such as Soar) produce cognitive behaviors, we need to insert information such as knowledge and rules into it. Thus, we developed our planning algorithms within the Soar architecture to let Soar devise a plan accordingly.

Figure 4 depicts pseudocode for the proposed planning algorithm for the evacuation application (see Section 2.1), which is implemented in Soar. Using this algorithm, individual agents (humans) develop their evacuation plans (routes) dynamically (involving from 1 to 10 planning steps) until they reach their destinations. The first line of Figure 4 shows that the application uses the BBN and DFT to obtain preference values for the paths directly accessible from the current position. Then, the algorithm works differently depending on the type of agent: novice or commuter. As noted in Section 2.5, the agent revises its plan when its *confidence index* falls below a threshold value. The following subsections discuss both cases in detail.

3.1 Multi-Horizon Planning Algorithm for Commuter Agent

As mentioned earlier, commuter agents represent people with enough knowledge about the area to plan beyond the current decision point (selecting a path from the current intersection to the next intersection). To illustrate the algorithm for various situations, an exemplary evacuation area (in Washington, D.C.) is used (see Figure 5 for satellite image and its corresponding graph). The graph (G) used here is defined formally as $G = (V, E)$, where V and E represent a set of nodes and edges pertaining to the graph, respectively. The graph (G) in Figure 5 has nodes $V(G) = \{a, b, c, d, e, f, g, h, I, j, k, l, m, n, o\}$ and edges $E(G) = \{ab, bc, be, de, ef, ej, fc, fg, gh, gk, hl, ij, jk, jm, kl, kn, lo, mn\}$.

In this example, it is supposed that an agent in node e is searching for a route (series of decisions, $R(G)$) to the destination node o. Figure 6 depicts a series of selection processes (of a path), which can be described as following:

(1) At node e (see Figure 6(a)), the agent evaluates each path (be, de, ef, ej)

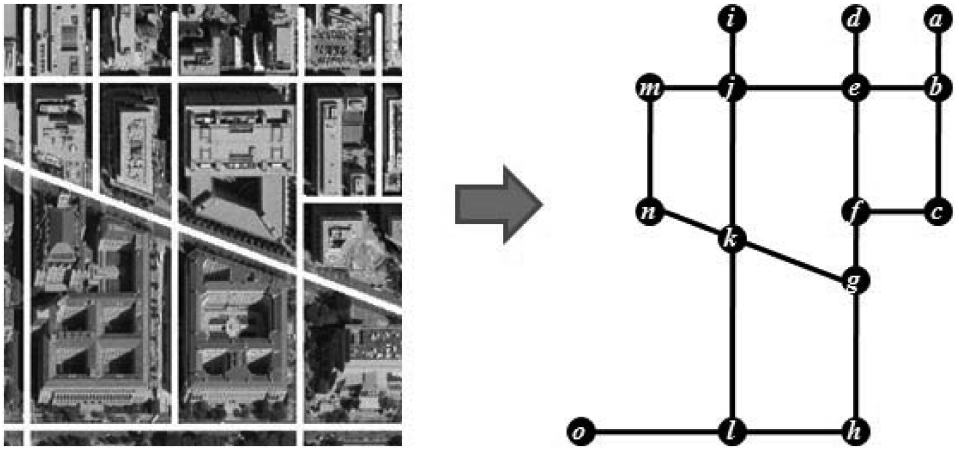


Fig. 5. Graph representation of the evacuation area.

in terms of smoke, fire, police, crowd, and the updated distance to the destination (distance to node o from nodes b, d, f, j).

(2) Based on his observations, the agent infers the evaluation matrix $M(t)$,

$$M(t) = \begin{pmatrix} m^{be}_{\text{risk}}(t) & m^{be}_{\text{time}}(t) \\ m^{de}_{\text{risk}}(t) & m^{de}_{\text{time}}(t) \\ m^{ef}_{\text{risk}}(t) & m^{ef}_{\text{time}}(t) \\ m^{ej}_{\text{risk}}(t) & m^{ej}_{\text{time}}(t) \end{pmatrix}$$

and weight vector $W(t)$,

$$W(t) = \begin{pmatrix} w_{\text{risk}}(t) \\ w_{\text{time}}(t) \end{pmatrix}$$

via BBN (see Figure 3)

$$M(t) = \begin{pmatrix} m(t)^{be}_{risk} & m(t)^{be}_{time} \\ m(t)^{de}_{risk} & m(t)^{de}_{time} \\ m(t)^{ef}_{risk} & m(t)^{ef}_{time} \\ m(t)^{ej}_{risk} & m(t)^{ej}_{time} \end{pmatrix}, W(t) = \begin{pmatrix} w(t)_{risk} \\ w(t)_{time} \end{pmatrix}$$

where $m(t)^{be}_{risk} m^{be}_{\text{risk}}(t)$ represents the evaluation of the *risk* attribute for edge *be* at time t, and $w_{risk}(t)w(t)_{risk}$ is the weight for the *risk* attribute at time t.

(3) The matrices $M(t)$ and $W(t)$ obtained in step 2 are provided to Eq. (1) (DFT), whose multiple replications generate the choice probabilities Pr(*be*), Pr(*de*), Pr(*ef*), Pr(*ej*) for each path *be, de, ef, ej*. For each replication, DFT evolution runs for enough time periods to allow convergence of the choice probability (see Section 2.4).

(4) Now, the choice probabilities are fed into Soar, and Soar selects one path randomly based on the probabilities (see Figure 6(b)). Suppose path *ej* is selected; then, $R(G)$ is updated to $\{ej\}$. Then, the same process (see

steps 1, 2, 3, and 4) is used to pick a second path from intersection j. The final step of the current iteration before starting the second iteration is to set the preference of ej (i.e., p_{ej}) to *worst* so that path ej (coming back to the intersection e again) will be elected only if there is no other choice in the second iteration. It is noted that while the second iteration of the planning algorithm starts from intersection j, the agent is still located at intersection e.

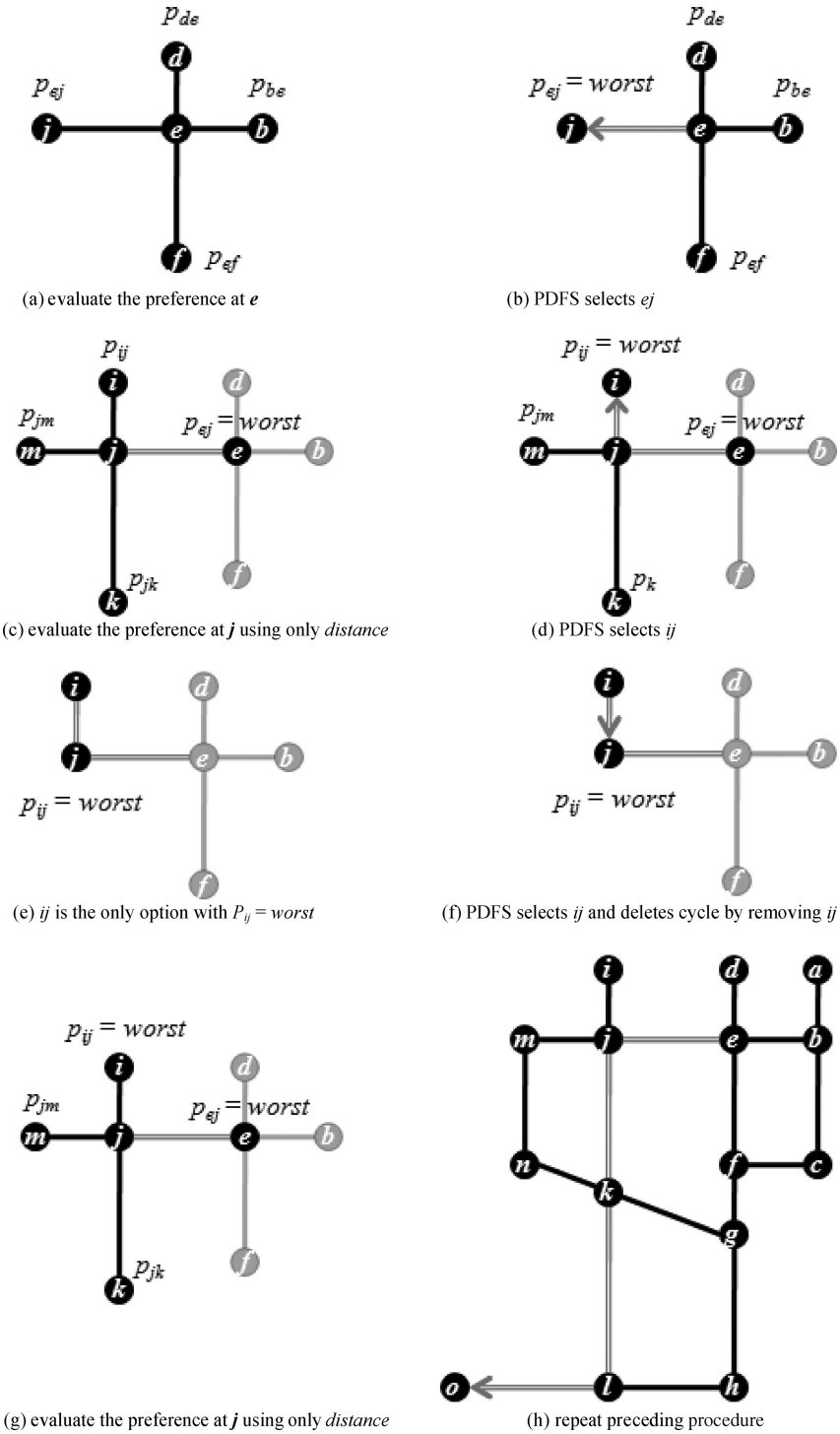
(5) At node j (see Figure 6(c)), the agent repeats step 1 to evaluate each path (ej, ij, jk, jm). However, this evaluation addresses only the updated distance to the destination (distance to node o from nodes e, i, k, m) as other environmental variables at intersection j (smoke, fire, police, and crowd) are not visible from the current location (intersection e) of the agent. Then, the agent repeats step 2 to infer the evaluation matrix $M(t)$ and the weight vector $W(t)$ via the BBN (see Figure 3), where the BBN uses the updated distance to the destination and the expected values for smoke, fire, police, and crowd. Then, the agent repeats step 3 to obtain the choice probabilities Pr(ej), Pr(ij), Pr(jk), Pr(jm), where the value (*worst*) of p_{ej} (see step 4) is not updated. Then, the agent repeats step 4, selecting path ij and updating $R(G) = \{ej, ij\}$ and $p_{ij} = worst$ (see Figure 6(d)). Note again that the agent is still planning the route without actually moving.

(6) At node i (see Figure 6(e)), the agent repeats steps 1, 2, 3, 4. However, as shown in Figure 6(e), the only available path is edge ij, whose p_{ij} was assigned as *worst* in step 5. In this case, although $p_{ij} = worst$ (implying a very small probability instead of zero probability), edge ij is selected and $R(G)$ is updated to $\{ej, ij, ij\}$ (see Figure 6(f)). Then, since edge ij has been taken twice, a cycle has been formed and the edges in the cycle are deleted from $R(G)$ (i.e., $R(G) = \{ej\}$) according to our planning algorithm. Then, Soar selects an edge from intersection j again based on the choice probabilities Pr(ej), Pr(ij), Pr(jk), Pr(jm), where both p_{ej} and $p_{ij}$are *worst* (see Figure 6(g)). This way, path ij is rarely selected again.

(7) The agent repeats the previous process until its plan reaches the destination node o or the limit of the planning horizon (n) is reached (see Figure 6(h)). In this multi-horizon planning process, the limit of the planning horizon is the same as the number of elements (edges) in $R(G)$. At this point, the multi-horizon planning process is complete, and the agent executes a decision based on the plan ($R(G)$).

3.2 Single-Horizon Planning Algorithm for Novice Agent

Novice agents (e.g., tourists) are those people who do not have knowledge of the area. Novice agents cannot develop a multi-horizon plan because they do not have any information (e.g., updated distance to the destination) other than what they see for the adjacent paths. Therefore, their planning horizon is one (see line 12 of Figure 4). It is noted that the planning procedure for the novice agent is the same as that of the commuter agent (steps 1 to 4 in Section 3.1), but with a planning horizon (n) of 1.



(a) evaluate the preference at ***e***

(b) PDFS selects *ej*

(c) evaluate the preference at ***j*** using only *distance*

(d) PDFS selects *ij*

(e) *ij* is the only option with $P_{ij} = worst$

(f) PDFS selects *ij* and deletes cycle by removing *ij*

(g) evaluate the preference at ***j*** using only *distance*

(h) repeat preceding procedure

Fig. 6. Illustration of planning algorithm.

3.3 Metamodel of Choice Probability for Commuter Agent

During the multi-horizon planning of the commuter agent (see Section 3.1), the choice probabilities for the paths (even beyond the current intersection) are calculated repeatedly via the BBN and DFT (see line 8 of Figure 4), which requires intensive computational power, especially when the simulation involves numerous agents. Thus, this section discusses an aggregated metamodel that allowed us to obtain the choice probabilities in a significantly shorter time. It is noted that both the original approach (the BBN-DFT method) and the metamodel can be used adaptively according to computational availability.

For planning from the current intersection to an adjacent one (all considered environmental variables are available to the agent), both the original approach (the BBN-DFT method) and the metamodel obtain choice probabilities using BBN and DFT in exactly the same way (see steps 1, 2, 3, 4 in Section 3.1). However, the original approach and the metamodel work differently for planning beyond the current intersection. The original (BBN-DFT) method uses only knowledge of the distance from the considered node to the destination to infer $M(t)$ (evacuation time and risk) via the BBN and to obtain the choice probabilities. On the other hand, the proposed metamodel uses knowledge about the number of connected paths from an intersection (which is related to risk) in addition to knowledge of distance (which is related to evacuation time). For example, considering four nodes *b, d, f, j* connected to node *e* in Figure 5, nodes b and f have three paths away from each of them, node d has one path away from it, and node j has four paths away from it. Here, we consider the number of connected edges emanating from a node because going to an intersection connected to more paths may be safer in an emergency evacuation situation. Based on knowledge about the distance from a connected to the destination and the number of connected paths emanating from that node, the preference value of each path is calculated using Eq. (5), as

$$p_i = k_i + \left(\mathrm{I}_{[\mathrm{x_{curr}},\mathrm{x_{dest}}]}(x_i) + \mathrm{I}_{[\mathrm{y_{curr}},\mathrm{y_{dest}}]}(y_i)\right) \cdot w_{\mathrm{dist}}, \tag{5}$$

where p_i = preference of path i; k_i = number of paths connected from the node connected to path i; I is the indicator function; $(x_i, y_i) = x, y$ coordinates of the node connected to path i; $(x_{\mathrm{curr}}, y_{\mathrm{curr}}) = x, y$ coordinates of the current node; $(x_{\mathrm{dest}}, y_{\mathrm{dest}}) = x, y$ coordinates of the destination node; w_{dist} = weight associated with the distance factor. The proposed metamodel accords a higher preference as the number of connected paths increases (less risk) or the distance to the destination decreases (shorter evacuation time). The weight associated with the distance factor (w_{dist}) allows us to adjust the impact of the distance and the number of connected paths. Once the calculated preference values are fed into Soar, Soar selects one path using the choice probabilities calculated from the preference values as described shortly.

To illustrate the proposed metamodel, we consider the same example used in Section 3.1 (searching for a route from node *e* to node *o*). For the decision from the current intersection (e) to an adjacent intersection, the agent uses the original (BBN-DFT) method. It is assumed that path *ej* is selected (i.e., $R(G) = \{ej\}$). This places the agent at node j, which is connected to nodes *e, i,*

k, m. The numbers of connected paths are $k_e = 4$, $k_i = 1$, $k_k = 4$, and $k_m = 2$, respectively, and the coordinates of the nodes are $(x_e, y_e) = (5, 7)$, $(x_i, y_i) = (3, 8)$, $(x_k, y_k) = (3, 4)$, and $(x_m, y_m) = (1.5, 7)$. It is assumed that the current node $((x_{\text{curr}}, y_{\text{curr}}) = (x_j, y_j))$ is located at (3, 7), the destination node $((x_{\text{dest}}, y_{\text{dest}}) = (x_o, y_o))$ is located at (0, 0), and w_{dist} is set to 2. Preference values for each node can be calculated using Eq. (5): (1) $p_e = 4{+}3/5{\cdot}2 = 5.2$, (2) $p_i = 1{+}7/8{\cdot}2 = 2.75$, (3) $p_k = 4{+}7/4{\cdot}2 = 6.8$, and (4) $p_m = 2{+}3/1.5{\cdot}2 = 6$. Here, as path ej already has been selected, p_e is set to *worst*. Next, Soar uses these preference values to calculate the choice probabilities: (1) $p_i = 2.75/(2.75 + 6.8 + 6) = 0.18$, (2) $p_k = 6.8/(2.75{+}6.8{+}6) = 0.44$, and (3) $p_k = 6/(2.75{+}6.8{+}6) = 0.38$, and selects a path randomly based on those choice probabilities. This planning procedure is repeated until it reaches destination node o or the planning horizon limit is reached. Comparisons between the original method and the metamodel (in terms of required computation and quality of generated plans) are left as a future research task, and the experimental results, which will be discussed in the next section, in the current work are based on the metamodel.

4. EXPERIMENT AND VALIDATION

This section describes a crowd simulation model that mimics the previously-described emergency evacuation scenario (see Section 2.1), utilizes human-in-the-loop experiments for human behavioral data collection, and facilitates testing the impact of several factors (e.g., demographics of agents, number of police officers, information sharing via speakers) on evacuation performance (e.g., average evacuation time, percentage of casualties). The crowd simulation model was developed based on the two-layer modeling principles proposed by Hamgami and Hirata [2003]: (1) modeling the agent and (2) modeling the environment that agents interact with, such as paths and intersections. By employing these two conceptual layers, we isolated models of the environment and agent, which made the modeling process easy. 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 perceptions of the environment and executes decisions to achieve its *intention* in the environment.

4.1 Simulation Model Development

The environment (path and intersection information representing the National Mall in Washington, D.C.) has been implemented in AnyLogic®6.0 agent-based simulation software. As discussed in Section 2, an agent plans and makes decisions via the BBN, DFT, and PDFS techniques. For implementation purposes, we employed various software packages: Netica for BBN, JAMA (a Java matrix package) for DFT, and Soar for PDFS. The fact that all of these software packages are Java based or have a Java interface has facilitated their integration. In our simulation, three types of agents are considered: (1) commuter, (2) novice, and (3) police agents. Each type of agent behaves differently. Commuter agents can be further defined as leaders, who lead the follower agents to the exits. And, both commuter and novice agents can be characterized as follower agents depending on the *confidence index*. When agents with a low *confidence index*

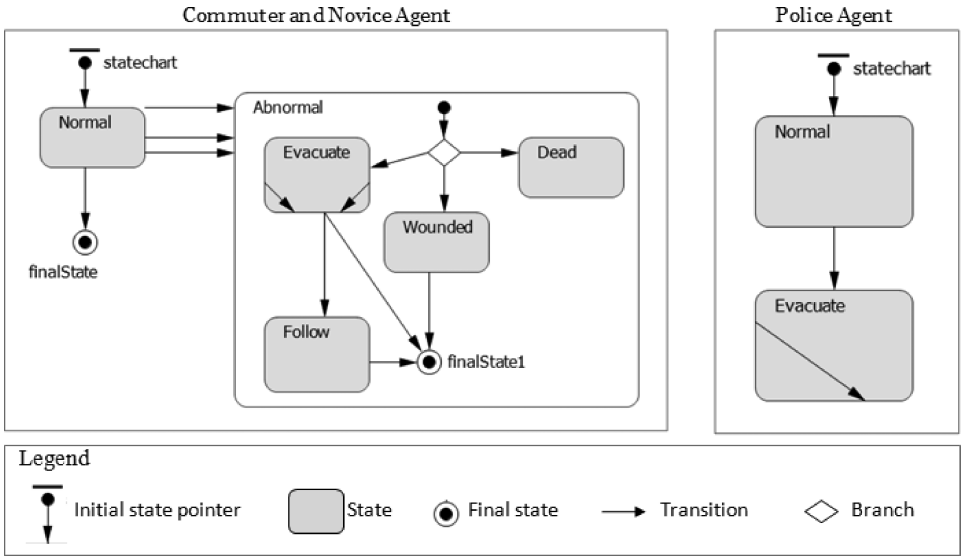


Fig. 7. State charts of the agent behaviors.

(follower agents) meet a leader agent, they start to follow that agent. Figure 7 depicts the behaviors of each agent using state charts. A state chart diagram is a generic way of representing the behavior of an agent in response to external events (e.g., an explosion) or internal events (e.g., achieving an intention). The *intention* of an agent is modeled in the state chart diagram through a sequence of transitions from one state to another. Every commuter and novice agent follows the state chart on the left, and every police agent follows the state chart on the right. Agents' individual behaviors are differentiated by parameters such as the *confidence index* and the planning horizon and by randomness within the BBN and DFT subroutines. The number of agents of each type in the simulation can be adjusted. As shown in Figure 7, when the explosion occurs, the commuter and novice agent's state transitions from "Normal" to "Abnormal" and the police agent's state changes to "Evacuate". When the commuter and novice agents transition into "Abnormal" state, their underlying state becomes "Evacuate", "Wounded", or "Dead" depending on their distance from the explosion. If a follower agent who is in the "Evacuate" state meets a leader agent, its state becomes "Follow".

Figure 8 depicts a snapshot of the AnyLogic®simulation, in which a bomb explosion is shown in the middle of the map and agents are evacuating the area. When the simulation begins, it generates the requested number of agents of each type and places them randomly within the selected area heading towards their everyday destinations (in the "Normal" state). Fifteen seconds after the simulation starts, an explosion occurs in the middle of the area. Based on distance from the explosion, agents within "fatal range" (as indicated by a smaller circle with solid line in Figure 8) of the explosion at that moment become dead. Similarly, the agents within "wound range" become wounded, and agents within "notice range" will notice the explosion and start to evacuate.

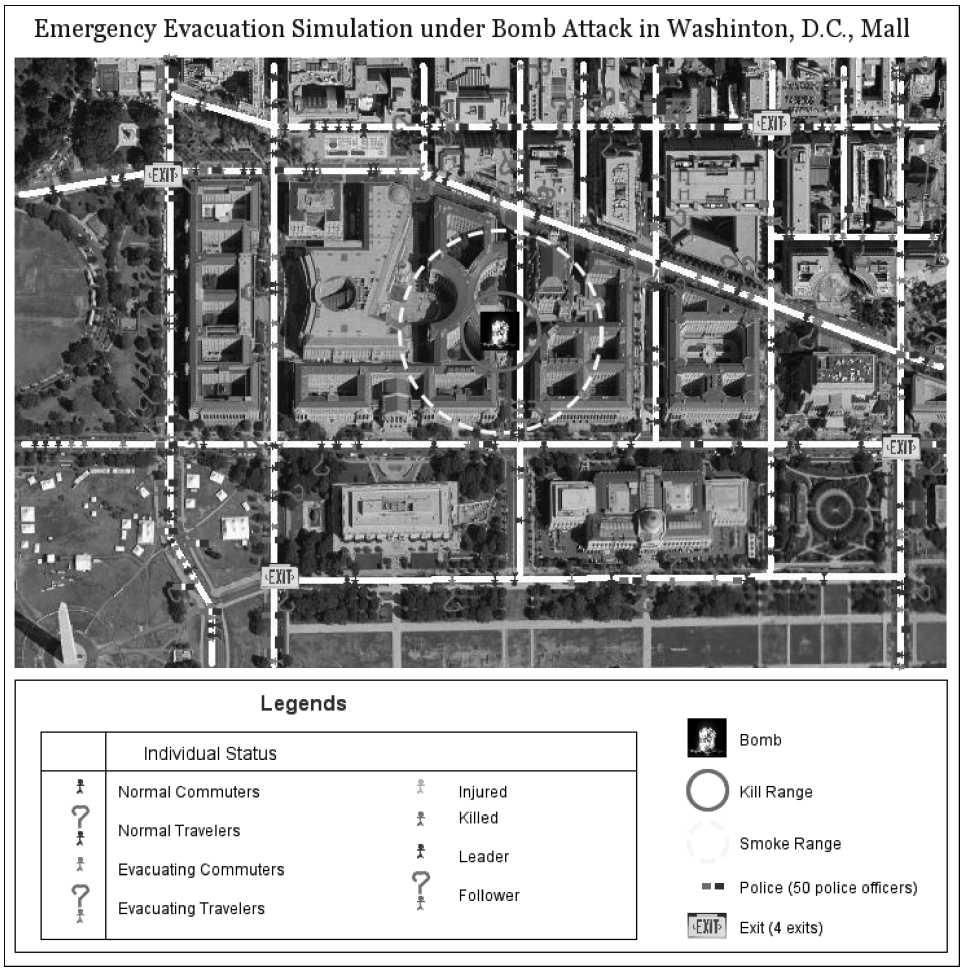


Fig. 8. Emergency evacuation simulation in AnyLogic®.

Smoke goes up and diffuses (as indicated by a dotted circle in Figure 8) from the explosion. In addition to the perception of sound and smoke, an agent can notice the explosion via communication with other agents and police. When two agents reach a certain proximity to each other, they can communicate and exchange information about the explosion. When agents notice the explosion, they start to move faster and head toward one of the four exits in the area. The simulation allowed us to observe the agents' behaviors that mimic humans in the given scenario; and using the simulation, we were able to evaluate various evacuation policies.

4.2 Virtual Reality Human Experiments

As discussed in Sections 2.2 and 2.3, the BBN infers $M(t)$ and $W(t)$, and DFT calculates preference values of the options based on those matrices of evaluations and weights. Thus, constructing an accurate BBN for a human

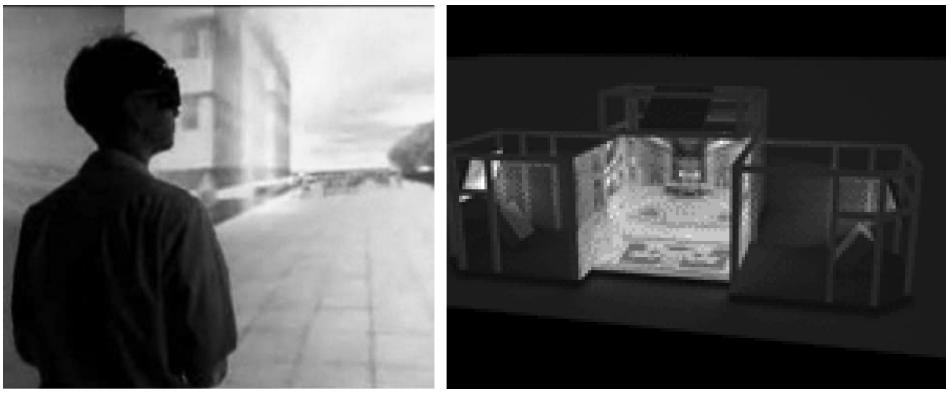


Fig. 9. Human-in-the-loop experiment and CAVE system.

is a critical task in the development of a simulation that accurately mimics human behavior. To this end, we conducted human experiments to extract their behaviors and translate those behaviors into a BBN. In this research, we used the Cave Automatic Virtual Environment (CAVE) to conduct human experiments in a 3D environment. "Immersiveness" means that the user's point of view or some part of the user's body is contained within the computer-generated space of the VR environment. Immersiveness allows us to observe quasireal human response data in a very practical way for a potentially life-threatening situation without actually putting humans at risk, whereas 2D-based experiments do not completely immerse the human subjects and would result in relatively unrepresentative participation from them [Shendarkar et al. 2006]. However, it is noted that other methods allowing subjects to imagine the scene such as text description, pictures, or movie clips may replace the CAVE system. The hardware system used is the FakeSpace, Inc., CAVE simulator. Figure 9 depicts a human-in-the-loop experiment in CAVE and the CAVE system, respectively. The 3D model projected within the CAVE system is developed using Google SketchUp 3D modeling software. The individual 3D images were collected from the Google SketchUp component library and the Google 3D Warehouse. Figure 10 depicts a snapshot of a virtual cityscape of an intersection developed by the Google SketchUp 3D modeling software.

In the human-in-the-loop experiment, each subject is asked to evaluate the risk and the evacuation time of 3 available paths (i.e., right, forward, and left) depending on the various environmental observations (i.e., fire, smoke, police, and crowd) at each intersection. Also, each subject is asked to select one of the 3 available paths. In this study, 6 volunteer subjects participated in the experiment. For each subject, an experiment involving 18 intersections with different environmental settings was repeated 3 times. The data collected on the relationship between the environment and the subject's evaluation was used to construct a BBN in the form of a conditional probability distribution.

Table I. Comparisons of Decisions between Human Subjects and Proposed Model

	Actual Decision			Simulation		
Intersection	Path1	Path2	Path3	Path1	Path2	Path3
1	0.15	0.75	0.1	0	0.97	0.03
2	0.4	0.2	0.4	0.75	0	0.25
3	0.3	0	0.7	0	0	1
4	0.05	0.55	0.4	0	0	1
5	0.35	0	0.65	0.41	0	0.59
6	0.35	0.1	0.55	0	0	1
7	0.15	0.3	0.55	0	0.14	0.86
8	0.6	0	0.4	0.21	0	0.79
9	0.78	0.17	0.05	1	0	0
10	0.333	0.333	0.333	0.18	0	0.82
11	0.5	0.11	0.39	0.1	0	0.9
12	0.61	0	0.39	1	0	0
13	0.11	0.83	0.06	0	1	0
14	0	0.89	0.11	0	1	0
15	0.11	0.89	0	0	1	0
16	0.11	0.89	0	0	1	0
17	0.83	0.06	0.11	0.51	0	0.49
18	0.22	0.72	0.06	0.07	0.93	0

4.3 Model Validation

In order to validate the proposed decision-making process (see Section 3.1) which is a part of real-time planning, we compared its results to the path selection decisions that humans made during the evacuation planning as shown in Table I. Each column of table i represents the probability that the path was selected by the human subjects (actual decisions from the human-in-the-loop experiment as described in Section 4.2) and the probability assigned to the path by the proposed model (simulation). For example, the first row of the Actual Decision column depicts that path1, path2, and path3 at intersection 1 were actually selected with probability 0.15, 0.75, and 0.1, respectively. if we compare the path having the highest probability in the simulation and the path selected most often by the subjects, the proposed model selected the same path as the subjects at 15 out of the 18 intersections. In other words, the proposed model selected the same decision as the actual human subjects about 83% of the time.

4.4 Simulation Results

Using the crowd simulation model that we constructed, we conducted various experiments to test the impacts of several factors (e.g., demographics, number of police officers, number of leader agents) on evacuation performance (e.g., average evacuation time). Figures 11 and 12 depict impacts of the number of police officers and the number of leaders, respectively. In Figure 11, the number of police officers increases from 10 to 100 with a step size of 10. In each simulation, 20 replications were conducted with 400 commuter agents and 100 novice agents. The average evacuation time of novice agents (travelers) was observed to decrease about 1 minute (25%) as the number of police

Fig. 10. VR model in Google SketchUp.

officers increased from 10 to 100. Reducing the average evacuation time by 1 minute in an emergency situation can save many lives. The average evacuation time of commuter agents, however, does not decrease much compared with that of the novice agents. This is because information from the police officers provides the travelers with new knowledge they did not have before, but it is only used by the commuters to complement/correct their knowledge/judgment.

Similarly, Figure 12 depicts the impact of the number of leaders (increased from 10 to 200 with a step size of 10) on average evacuation time. Each simulation, involving 400 commuter agents and 100 novice agents, was replicated 20 times. In order to eliminate the police effect in this experiment, the agents are configured to notice the explosion right after it occurs and start to evacuate. This configuration results in smaller evacuation times compared with the results in Figure 11. The results in Figure 12 reveal that an increase in the number of leaders from 10 to 200 reduces the agents' evacuation time an average of about 30 seconds, which is critical in an emergency situation. In addition, the results are consistent with our intuition that commuters have less dependency on leaders than do travelers, as they usually have a higher *confidence index* than travelers, and agents start to act as followers if their *confidence index* is low. It is noted that the simulation is flexible, so it can be used to test impacts of other factors (e.g., impact of information sharing via

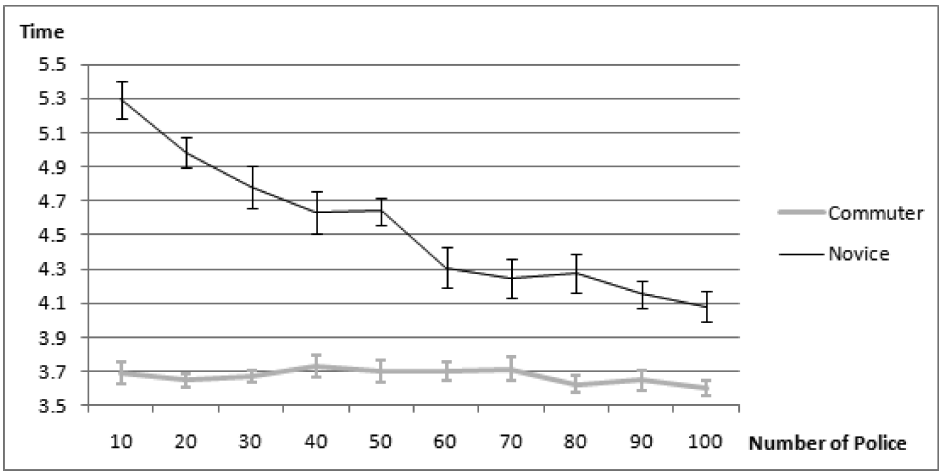


Fig. 11. Impact of number of police officers on the average evacuation time and 95% confidence interval.

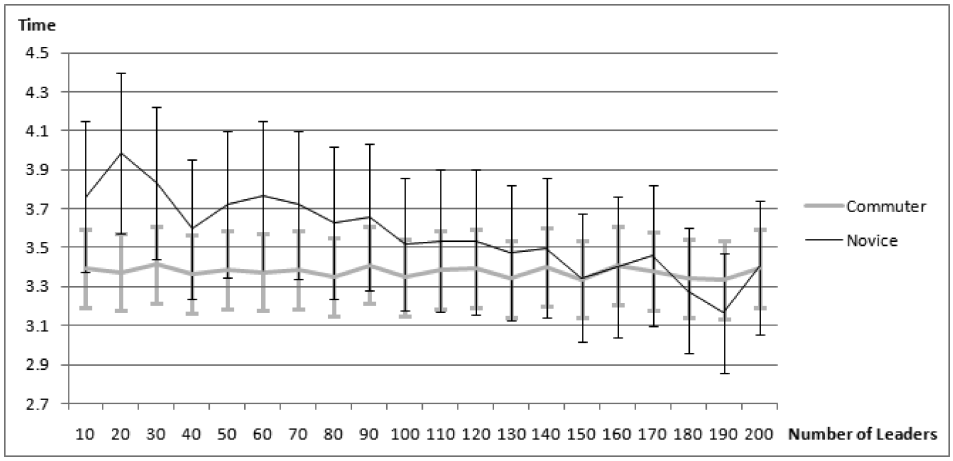


Fig. 12. Impact of number of leaders on the average evacuation time and 95% confidence interval.

speakers or text messaging) on various other security metrics (e.g., percentage of casualties).

5. CONCLUSION AND FUTURE RESEARCH

In this research, we proposed promising techniques to realize each submodule of the extended BDI architecture. The techniques employed in this research were selected to represent the characteristics of the corresponding steps of the human decision-planning and decision-making process under selected emergency evacuation scenarios. Successful implementation of these techniques allowed the extended BDI architecture to be used to mimic human behaviors even in complex situations. Furthermore, the proposed techniques and the extended BDI framework were demonstrated using agent-based simulation,

which provided various dynamic environments. The simulation we developed allowed us to simulate and observe crowd behaviors under evacuation scenarios with various conditions. The proposed simulation has the potential to allow the responsible governmental and law-enforcement agencies to evaluate different evacuation and damage control policies beforehand, which in turn would help them to execute the most effective crowd evacuation scheme during an actual emergency situation. As part of this research, we conducted human-in-the-loop experiments using a virtual reality system to collect data regarding more realistic human behaviors. Currently, the learning effect of agents is not considered but is left as future work. Through the learning process, we can represent how behaviors of a novice agent become closer to those of commuter agents.

REFERENCES

BUSEMEYER, J. R., AND DIEDERICH, A. 2002. Survey of decision field theory. *Math. Social Sci. 43*, 345–370.

BUSEMEYER, J.R., AND TOWNSEND, J.T. 1993. Decision field theory: A dynamic-cognitive approach to decision making in an uncertain environment. *Psychol. Rev. 100*, 3, 432–459.

EARNSHAW, R.A., VINCE J.A, AND JONES H. 1995. *Virtual Reality Applications*. Academic Press Ltd., London, UK.

EDWARDS, W. 1954. The theory of decision making. *Psychol. Bull. 51*, 4, 380–417.

EINHORN, H.J. 1970. The use of nonlinear, noncompensatory models in decision making. *Psychol. Bull. 73*, 3, 221–230.

GAO, J., AND LEE, J.D. 2006. Extending the decision field theory to model operator's reliance on automation in supervisory control situations. *IEEE Trans. Syst. Man Cybernet. Part A 36*, 5, 943–959.

GIBSON, F.P., FICHMAN, M., AND PLAUT, D.C. 1997. Learning in dynamic decision tasks: Computational model and empirical evidence. *Organiz. Behav. Hum. Decis. Process. 71*, 1–35.

GLIMCHER, P.W. 2003. *Decision, Uncertainty, and the Brain, The Science of Neuroeconomics*. MIT Press, Cambridge, MA.

HAMAGAMI T., AND HIRATA H. 2003. Method of crowd simulation by using multiagent on cellular automata. In *Proceedings of IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)*. 46–52.

KINNY, D., GEORGEFF, M., AND RAO, A. 1996. A methodology and modeling technique for systems of BDI agents. In *Proceedings of the 7th European Workshop on Modeling Autonomous Agents in a Multi-Agent World MAAMAW'96*. W. Van Der Velde AND J.W. Perram, Eds. Springer Verlag, 56–71.

KONAR, A., AND CHAKRABORTY, U.K. 2005. Reasoning and unsupervised learning in a fuzzy cognitive map. *Infor. Sci. 170*, 419–441.

LAIRD, J.E., NEWELL, A., AND ROSENBLOOM, P.S. 1987. Soar: An architecture for general intelligence. *Artif. Intell. 33*, 1–64.

LEE, S., SON, Y., AND JIN, J. 2008. Decision field theory extensions for behavior modeling in dynamic environment using Bayesian belief network. *Infor. Sci. 178*, 10, 2297–2314.

MOSTELLER, F., AND NOGEE, P. 1951. An experimental measurement of utility. *J. Polit. Econ. 59*, 371–404.

NEWELL, A. 1990. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA.

NORLING, E. 2004. Folk psychology for human modeling: Extending the BDI paradigm. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent System*, 202–209.

OPALUCH, J.J., AND SEGERSON, K. 1989. Rational roots of irrational behavior: New theories of economic decision-making. *Northeastern J. Agricul. Resource Econ. 18*, 2, 81–95.

PAYNE, J.W. 1982. Contingent decision behavior. *Psychol. Bull. 92*, 382–402.

RAO, A.S., AND GEORGEFF, M.P. 1998. Decision procedures for BDI logics. *J. Logic Comput. 8*, 3, 293–343.

ROTHROCK, L., AND YIN, J. 2008. Integrating compensatory and noncompensatory decision making strategies in dynamic task environments. In *Decision Modeling and Behavior in Uncertain and Complex Environments*, T. Kugler et al., Eds. Springer, 123–138.

SANFEY, A.G., LOEWENSTEIN, G., MCCLURE, S.M., AND COHEN, J.D. 2006. Neuroeconomics: Cross-Currents in research on decision-making. *TRENDS Cogn. Sci. 10*, 3, 108–116.

SEN, S., ASKIN, R., BAHILL, T., JIN, J., SMITH, C., SON, Y., AND SZIDAROVSZKY, F. 2008. Predicting and prescribing human decision making under uncertain and complex scenarios. MURI (award number: F49620-03-1-0377) 2007 annual report.

SHENDARKAR, A., VASUDEVAN, K., LEE, S., AND SON, Y. 2006. Crowd simulation for emergency response using BDI agents based on immersive virtual reality. *Simul. Model. Prac. Theory 16*, 1415–1429.

SHIZGAL, P. 1997. Neural basis of utility estimation. *Current Opin. Neurobiol. 7*, 198–208.

SIMON, H.A. 1955. A behavioral model of rational choice. *The Quar. J. Econ. 69*, 99–118.

SIRBILADZE, G., AND GACHECHILADZE, T. 2005. Restored fuzzy measures in expert decision-making. *Inf. Sci. 169*, 71–95.

ZHAO, X., AND SON, Y. 2008. BDI-based human decision-making model in automated manufacturing systems. *Int. J. Mode. Simul. 28*, 3, 347–356.

Received April 2008; revised December 2009; accepted December 2009