Simulation Modelling Practice and Theory 85 (2018) 15–32

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

A simulation framework for crisis management: Design and use

Seth N. Hetu, Samarth Gupta, Vinh-An Vu, Gary Tan*

National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 14 November 2017
Revised 28 February 2018
Accepted 3 March 2018

Keywords:
Crisis management simulation
Symbiotic simulation

ABSTRACT

A perennial simulation framework is proposed within the domain of crisis management simulation. Motivated by a need for establishing information superiority through decision-support analysis, the framework is designed to use symbiotic simulation and is also suitable for the hindsight and foresight studies that drive crisis-related preparedness exercises. The framework provides a novel feature of incorporating *Human in the Loop* simulations using virtual reality as a part of the symbiotic simulation. We coin the term perennial simulation to refer to our framework being *enduring* (the simulation working symbiotically with the real system), and *recurring* (performing "what-if?" simulations and continually providing feedback to the real system).

Three case studies examine the application of the framework to crisis-related scenarios. The framework is shown to be useful and capable of dealing with crisis situations and adding value to existing expert advice, forming a symbiotic feedback loop that aids crisis management.

1. Introduction

The magnitude of preparedness in crisis management is not to be understated. The nature of a crisis is unpredictable and rare, but when similar crises occur, a "before and after" comparison of management techniques is striking. Two hurricanes of similar intensity struck Galveston and Texas at the turn of the 20th century, but the latter claimed 99.1% fewer lives [1,2]. Sometimes, comprehensive preparedness reaps additional rewards. Motivated by concerns regarding a critical influx of people into the city of Los Angeles for the 1984 summer Olympic games, officials installed a modern, automated traffic control system that exists to this day [3,4]. Such an event is termed "crisis-similar", as it has similar unpredictability, mitigation techniques, and focus on preparedness as more well-known crises such as natural disasters. The common constraints on all crises are information and time, with the crisis manager attempting to achieve information superiority over a crisis immediately after its occurrence [5,6]. With these observations in mind, we present a perennial simulation framework for management of crises and crisis-similar events. In addition, our framework attempts to exploit modern technological developments in the field of computer science. The framework is also flexible enough to provide hindsight and foresight studies for use in crisis-related preparedness exercises. We call our framework perennial because it is enduring and recurring - enduring because the computer simulation works symbiotically with the real system it is simulating, and recurring because the framework continually performs "What-if?" simulations to source for the optimal (or sub-optimal solution) for

* Corresponding author.
E-mail address: gtan@comp.nus.edu.sg (G. Tan).

https://doi.org/10.1016/j.simpat.2018.03.001
1569-190X/

feedback control into the real system. The framework also allows humans in the loop (HITL) to run in the symbiotic environment. To our knowledge, this has not been applied before in symbiotic simulation. In addition to the contribution of the framework itself, we also provide a proper categorization and classification of different types of crises. Following this, we describe several real-world case studies to test the efficacy of the proposed framework. These contribute to the knowledge and breadth of the inter-disciplinary field of crisis management simulation. The remainder of this paper is organized as follows. Section 2 covers related work in the field of crisis management simulation and how our framework compares with similar work in symbiotic simulation and dynamic data-driven application systems (DDDAS). Section 3 describes the framework. Section 4 covers its usage in three case studies, together with the results analysis. Section 5 concludes the paper.

2. Literature review

2.1. Definitions and taxonomy

The notion of *crisis management* is dependent upon an event known as a *crisis.*[1] Within this paper, these terms are defined as follows. These terms have not been standardized within the crisis management community, but our provided definitions are consistent with their usage in the literature.

Definition 2.1 (Crisis)**.** A crisis is a disruptive event that is difficult to predict. If mismanaged or otherwise left unchecked, a crisis will have a cascading effect, leading to a loss of life or resources. Crises may either occur naturally or be instigated and exacerbated by an iniquitous entity. Crises require swift action to mitigate their destructive potential.

Definition 2.2 (Crisis management)**.** Crisis management is the set of direct actions taken to prepare for, respond to, and mitigate a crisis event. These actions involve intervention points between various escalating stages of a crisis, and use information superiority in an attempt to disrupt the cascading effect of a crisis.

The taxonomy of crises is broken down into three categories. Most early work considers natural or man-made *accidental disasters* such as fires, earthquakes, and major traffic collisions [5–8]. The second general category is an *incited incident*, which includes agents acting against the goodwill of the general public [7,9]. These include terrorist attacks and insider threats, and are complicated by the fact that the instigator of the crisis continues to consciously act to prolong the crisis. Finally, a *business disaster* centres around loss of capital or public confidence for a given corporation. These types of crises may involve insider trading and poisoned product scares, but are not considered within the scope of our framework.

All crises share similar properties. The most prominent characteristic of a crisis is its rapid scaling-up of small-scale disasters into wider, full-blown moments of panic [10]. This reflects the disruptive and cascading nature of a crisis. The second most prominent characteristic of a crisis is the use of information sharing to counter the compounding escalation of such events [11]. A crisis is difficult to predict, but quick and informed action can drastically reduce its effect.

2.2. Related work

This section covers related work in crisis management, its simulation and other relevant fields. This work is split between understanding the physical event that causes the crisis, understanding how to react to it, and developing the tools needed to accomplish this.

A great deal of work has been done to understand the causes of various crises. Fire spread models are thoroughly understood in most interior and exterior contexts [12–17]. Floods and earthquakes have also been heavily researched. Recently, studies of infectious diseases have become more frequent. Affected zones ranging in size from a single university campus to an entire urban region are analyzed to determine the spreading patterns of viruses such as influenza, bird flu, or HIV [18–20]. Healthcare uses simulation to avoid crises such as hospital overload. As hospitals operate near peak capacity, priority inversions and patient no-shows can, if unchecked, lead to overload. [21] attempts to reduce the former through the use of “Provider-Directed Queueing”, with improvements in waiting time of 44% to 76%. [22] attempts to solve the latter by reducing the number of appointment *types*. Such analytical approaches are reminiscent of crisis management’s focus on information superiority over a crisis.

At a more general level, several researchers focus on secondary tasks common to all crises. Studies of infectious diseases tend to focus on preparedness, speculating, for example, on the effect that various historical outbreaks would have in a modern urban environment [19]. A fire crisis management system, on the other hand, focuses on the impact and the evolution of the accident in order to determine optimum routing plans which minimize the routing time for emergency vehicles to move and for people to leave the area [15]. An earthquake can be studied from the perspective of the resources allocated immediately after the event [23]. Bombings can also be understood from the point of view of the egress directly after the attack, rather than the physical explosion itself [24]. Studies like these rely on the assumption that allocation and movement of human beings or resources is critical the closer one gets to the moment of crisis. The “prioritization of limited resources” has been identified by the World Health Organization as a key goal in dealing with infectious diseases [25]. Other tools

[1] For the remainder of this paper, we consider “crisis-similar” events to be equivalent to crises.

used to this effect include High-Level Architecture-based crowd movement simulations and Nash equilibriums for resource management [10,26].

Regarding the tools for crisis management, the use of decision support systems has rapidly increased. This has strengthened capacity to prevent and suppress the crises while protecting human lives and property. According to Sakellariou et al. [27], most decision support systems for managing forest fire (and other crises) comprise the following components: (1) Retrieval, analysis, update, edit and prediction models of data, (2) Risk indexes and maps based on past incidents, (3) Crisis propagation and behavior models and (4) Interactive programs for the preparation, planning, coordination and prompt dispatch of rescue teams. A recent trend for crises–related decision support systems is to make use of crowd-sourced data and/or real–time messages shared on the popular social networks, and then applying the data mining techniques to identify the location and the consequences of the events and assist in mitigating the impact of the crisis [28,29].

From the modelling & simulation perspective, different models have been proposed for crisis management using agent based modelling and explaining crisis as a business process. Martagan et al. [30] proposes a simulation model for port operations during crisis conditions, demonstrating how simulation can be an effective tool to mitigate the effect of crisis on the performance of supply chains. An agent based model for an organisation has been developed in [31], comprising three layers of abstraction: Service Layer, Coordination Layer and Organization Layer. It supports exploring new scenarios while maximising the re-usability of existing agents. Dihé et al. [32] proposes a modular framework with main focus on planning and decision support systems for modelling and simulation of realistic crisis scenarios. For validation, five pilot sites have been used across a wide range of crisis management situations. Using a multi-agent co-operational model [33] explains crisis management as a three-step business process. Gonzalez [34] combines discrete event simulation (DES) and agent based modelling (easily extended to multi-agent simulation (MAS)) into a single crisis response model architecture. This kind of architecture, in which both approaches are combined, allows for different combinations of crisis response organizations (MAS-based) against different crisis scenarios (DES-based).

2.3. A comparative analysis - Symbiotic, DDDAS & HITL

Symbiotic simulation describes a paradigm in which a simulation system and real physical system are closely associated with each other. The simulation system is continuously executing and attempts to optimise the physical system in a way that is mutually beneficial. The simulation system benefits from sensor data while the physical system benefits from decisions made based on the simulation results [35]. Based on the sensor data, the simulation system performs multiple “What-if?” simulations, the results of which are analysed and used to predict or control the future behaviour of the physical system. Symbiotic simulation can be applied to many application domains: semiconductor manufacturing [36], and path planning for unmanned aerial vehicles [37].

Symbiotic simulation adds a feedback loop to a traditional simulation, allowing for critical events to be predicted in advance and averted. Current work in symbiotic simulation focuses more on regimented domains, such as manufacturing, shipping, or housing [38–40]. Although the nature of crises usually precludes prediction, many of its escalating factors (e.g., overcrowding, resource scarcity) can often be intercepted early. The field of agent-based simulation is also relevant. Here, autonomous reacting entities called agents are used to model human behaviour in an abstract way. This may be used, for example, to attempt to recreate optimistic bias in games of chance [41]. An overview of common agent-based simulation approaches is provided by Lee and Son [42]. Combinations of the two approaches already exist. [43] connects an agent-based evacuation model to a fire-spread model to capture the effects of egress behaviour on fire spread. Agents in this simulation can break windows and open doors in an attempt to escape a burning building, both of which will affect air currents and thus change the overall spread of the fire. CIPRTrainer [44] is another application that provides a new capability for training crisis management staff using “What-if?” simulations. It enables exploring different courses of action and comparing their consequences (what-if analysis) in complex simulated crisis and emergency scenarios.

A field of study closely related to symbiotic simulation is dynamic data-driven application systems (DDDAS) [45]. A dynamic data-driven application system is a paradigm whereby the computation and instrumentation aspects of an application system are dynamically integrated in a feedback control loop, where the instrumentation data can be dynamically incorporated into the executing model of the application, and in turn, the execution model can control the instrumentation. A summary of the differences and similarities of symbiotic simulation and DDDAS can be found in [46].

Human in the Loop (HITL) simulation is another field of study which features more prominently in virtual environments [47], and is used in training exercises to improve the cognitive abilities of the trainee. In this type of simulation virtual reality plays a major role. The most relevant and widely used software in this case is Cave Automatic Virtual Environment (CAVE), a fully immersive 3D virtual environment that is occasionally used to extract or validate agent models [42]. In our framework, we present a technique called Massively Multiplayer Online Human-In-the-Loop Simulation, or MMOHILS, that attempts to make virtual reality a feature of our framework [48,49]. This is done by incorporating existing techniques from the field of Massively Multiplayer Online games.

Table 1shows a comparison of the proposed perennial framework with existing common simulation techniques such as symbiotic and DDDAS (adapted from [46]). The novelty of the framework lies in its approach to incorporate the existing techniques along with the HITL simulation as sub-parts of a single integrated multi-purpose framework. This framework is validated by three case studies in Section 4. A frequently encountered situation when the simulation systems are interfaced with real world is that of noise and scarcity of data obtained from sensors. This problem is further elevated in the case of

Table 1
Comparison between our proposed framework and other existing techniques

Paradigm	Steering of measurement process	Control feedback	Data-driven application/simulation	What-if analysis	Human in the loop
Symbiotic	Optional	Optional	Simulation only	Yes	No
DDDAS	Mandatory	Optional	Both	Optional	No
Perennial	No	Yes	Both	Yes	Yes

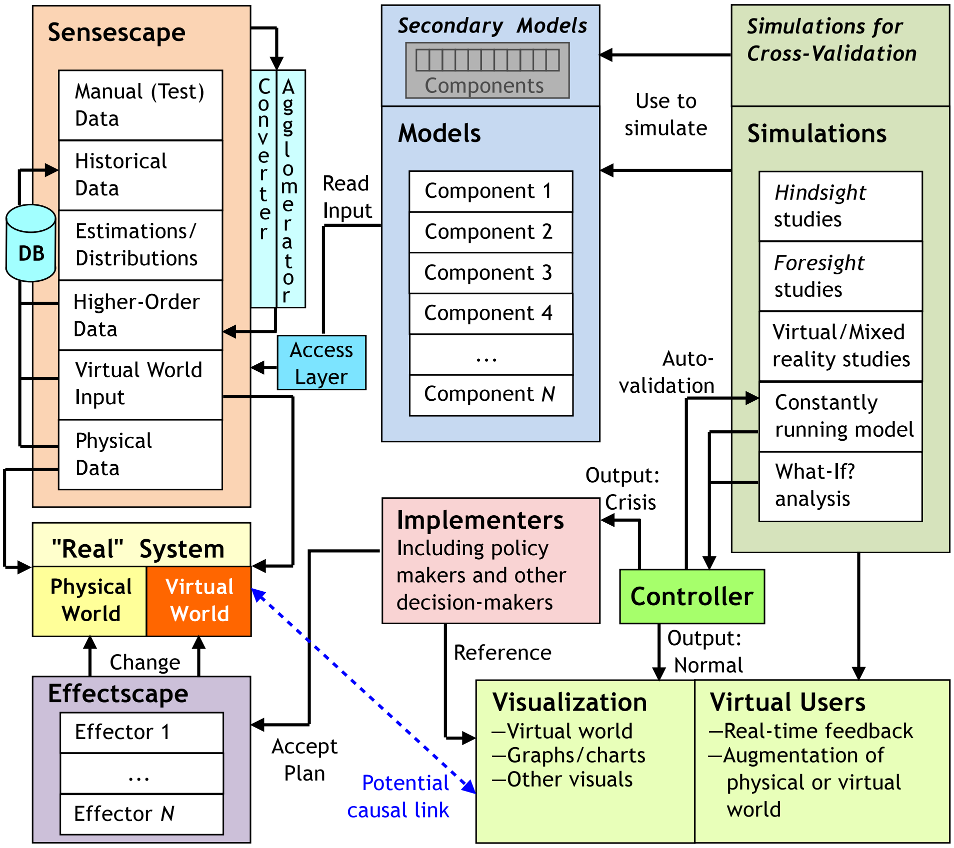


Fig. 1. Perennial Simulation framework organization.

crisis where sensor data may not available or the reliability of available data cannot be established with high confidence. This is a serious limitation especially in the case of dynamic data driven applications. In such situations, our proposed perennial framework can be a good substitute. Case study 3 in Section 4 uses a closed loop approach to deal with such situations where two different simulators run in a symbiotic relationship while only one has been calibrated with real world historical data and hence acts as a surrogate for real world or a physical system.

3. Proposed framework

A framework for constructing perennial simulations was designed, and a lightweight implementation was realized [50]. The previous section established the research community's need for an enduring system which is robust for a myriad variety of studies; this is what we have termed the perennial simulation framework. This framework is structured as a series of atomic components that extend their traditional simulation counterparts to provide perennial simulation functionality through composition. This section will elaborate on the proposed framework and its intended usage. The framework mandates a series of interfaces (typically as Java abstract classes) that a given perennial simulation extends to meet its specific requirements.

The components and interaction of a perennial simulation are detailed in Fig. 1. The primary relationship is between the **Real System**, which includes relevant physical and virtual locations of interest, and the **Implementers** or **Controller**, who are attempting to study some aspect of these locations. The Controller typically manages foresight and hindsight-driven studies, while the Implementers are characterized by their need for symbiotic decision support. The remaining framework components exist to enhance usage of the system as a whole [51].

Table 2
Properties of sensors and effectors

Optional	Property (Key)	Possible Values	Purpose
N	**name**	(*identifier*)	Uniquely identifies this sensor or effector. (E.g., "*sensor.water-level*")
N	**target**	(*identifier*)	Specify the modeled construct being sensed or changed.
N	**world**	*physical.X virtual.Y*	Denotes which world contains the target.
Y	**range**	(*text*)	Range of values measurable by sensors; range of values effectors accept as input.
N[a]	**valid- datetime**	(*Start, End*), *Infinity*, *Bound* + *Real-time?*	Specifies the range this data is valid for. "Infinite" data can be generated. "Real-time" is an optional flag, and means that data will become available later; bound sensors rely on other sensors' *valid-datetime*.
Y[a]	**mapping- function**	$S = \{s1, s2, s3\}$ $F(S) \rightarrow value$	Set of sensors, and a function which maps these sensors to a combined value.
Y[b]	**restricted**	*true false*	Whether implementer approval is required to activate actuation.
Y	**...**	(*Varies*)	Additional, user-defined properties

[a] only applies to sensors
[b] only applies to effectors

The subset of the real system which can be monitored by sensors forms the **Sensescape**, while the analogous concept for effectors is the **Effectscape**. Individually, **sensors** read data from the real system, while **effectors** are used to change the system based on the Implementers' decisions. Properties of sensors and effectors are listed in Table 2. The **name** is used to identify the sensor, while the **world** and **target** disambiguate the specific location being sensed. Sensors will not generate data outside their **range**, and an effector will only accept input that falls within this value. The **valid-date-time** similarly restricts the domain of a sensor.

The **mapping-function** is a sensor-only property that is key to establishing a diversity of sensor types. It consists of a lambda-calculus function —which may be realized programmatically— that combines the input of several sensors or converts raw sensor data into a higher-level semantic format ("Agglomerator" and "Converter" shown in Fig. 1). The **restricted** flag is used to limit sensor access to Implementers in the event of a decision-support study. Finally, sensors may have additional user-defined functionality.

The **Models** and **Simulations** serve to simulate the crisis event itself and any supporting data, while the various **Validation** studies serve to ensure the entire system is correct within its original assumptions. Categorically, all of the simulations in the system fall into one of two groups: **primary simulations**, which represent essential studies, and **secondary simulations**, which exist for a variety of miscellaneous purposes. All **primary simulations** are *agent-based* simulations. Agent-based simulations encapsulate behaviour into entities which sense and modify their environment, and communicate with each other [52]. This technique has been shown to model complex human behaviour in a fairly intuitive manner, and this intuitiveness can ease certain aspects of validation.

Human behaviour models are necessary in most crisis management studies, as crises tend to feature a strong human component, and require more than simple physics-based models to capture the variability of human decision-making.

The **primary simulations** will generally belong to one of several categories. Most simulations will exist for hindsight, foresight, and "What-If?" guided symbiotic decision support. In addition, a constantly-running model is also needed both for auto-validation, and to trigger the "What-If?" analysis when a thresh-hold has been crossed or becomes imminent. Primary simulations will typically rely on **primary models**. The **secondary simulations** and other secondary components are designed to encapsulate so-called "legacy simulations".

A legacy simulation typically is not agent-based and relies on **secondary models** that do not interact easily with the remainder of the framework. Despite their reduced flexibility, legacy simulations are a useful transitional technology for including existing (non-perennial) simulations in our framework. Although not specified by the framework, the primary simulations are typically stochastic in nature. This allows them to fully explore the nature of the "What-If?" search space, which can be affected by emergent effects and real-world randomness.

Our framework also supports the running of **virtual** or **mixed reality studies**, which are emerging as an effective way of gathering more realistic data in environments that are too costly or dangerous to explore otherwise. A number of **Visualization** components present the virtual worlds to any human-in-the-loop participants, as well as providing accumulated statistics to the Implementers. Several additional components, such as **DB Access** and the **Converter/Agglomerator**, are of minor importance and described fully in [50].

4. Case studies

Using the framework as described in previous section, three case studies were devised to test its applicability in real-world crisis management studies. The first of these was modelled for a building egress scenario, and tests the effect of symbiotic feedback using humans in the loop on a systematic evacuation. The second experiment was inspired by the

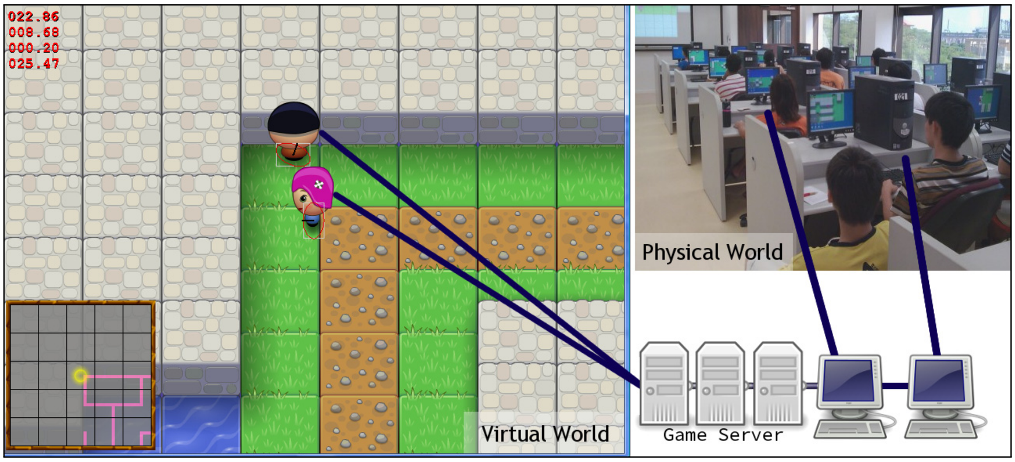


Fig. 2. Two players exploring an online virtual world.

previously-discussed traffic studies, and attempts to provide advanced information to incident response teams on an arterial road network. The third case study for traffic management using guidance is carried out to show that the system is robust enough to handle noisy or sparse sensor data. These experiments together exercise a substantial portion of the framework.

4.1. Guided egress scenario using human in the loop

For the first study, a pedestrian movement MMOHILS was created and used for symbiotic-guided evacuation of a virtual building environment undergoing a mild hypothetical crisis. This "library egress study" was a complete, self-contained simulation study, and was the inspiration behind the perennial framework as a whole [49]. In this case study, real users partake in the library egress in a virtual world and are represented by avatars in the world. The library egress study was performed to assess the effect of introducing a symbiotic feedback loop into a simulation of pedestrian egress. The benefit of symbiotic feedback was measured by comparing an egress simulation with and without symbiotic feedback. We expected that symbiotic feedback would decrease both total evacuation time and average evacuation time, as well as improving the cohesiveness of egress on the whole.

4.1.1. Experimental setup

A virtual world was created from existing university library blueprints. Each floor measured 45.7 m by 29.2 m, with actual walkable space limited to 983.09 m^2 on the first floor, and 651.98 m^2 on the second floor. The virtual environment was then scaled to the size of the users, based on a bird's-eye view common to most pedestrian experiments [53–56]. In particular, Fruin [57] models pedestrians as ellipses viewed from a top-down perspective. Hence, our participants' virtual avatars were similarly represented. Participants use intuitive mouse-centric controls to move around by clicking in the direction they wish to travel. Each avatar has a heading which rotates to match the direction of movement. Avatars collide with environmental obstacles and with each other, which halts their forward movement. Visual consistency is maintained by always showing the current player's avatar with pink hair; unintentional collusion is mitigated by portraying all other avatars with identical looking brown-haired models. Fig. 2 depicts several real-world participants progressing and navigating through an arbitrary "warm-up" virtual world.

At the time of the simulated egress event, all agents are dispersed among the second storey book stacks. From here, they must proceed to one of four staircases and descend to the first floor. The first floor has one main exit in the front of the building and a second, smaller exit near the back. The stated cause of the evacuation was a non-dangerous ceiling tile collapse on the first floor. This served a dual purpose: it allowed us to introduce bottlenecks (i.e., debris from the collapsed tiles), and it provided an impetus for evacuation that did not also imply peril. Dangerous evacuation scenarios such as fires or earthquakes would have required more rigorous validation of each user's psycho-social state, which was beyond the scope of our experiment.

In addition to the constantly-running model (which approximated an agent-based system), an egress model is triggered at various intervals to perform "What-If?" analysis of the building's crowd movement patterns. Egress was treated as a network flow problem, and modelled using the fast and powerful EvacNET modelling software. EvacNET treats agent egress as a network flow problem, and was calibrated to Fruin's measurements of pedestrian capacities [57]. EvacNET does no behavioural modelling, so it always arrives at an optimal egress strategy. This strategy was communicated to users, forming a symbiotic feedback loop. The end goal was to guide users optimistically towards a "best case" egress scenario. Figs. 3 and

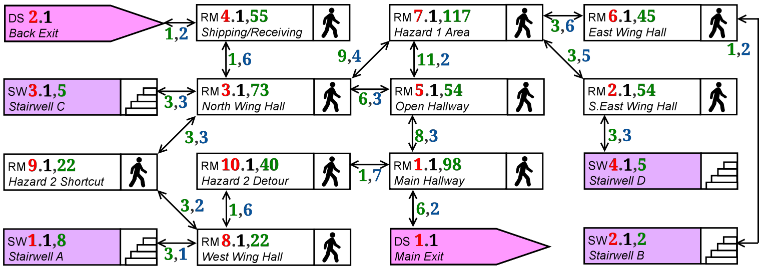


Fig. 3. EvacNET model of the library, first floor.

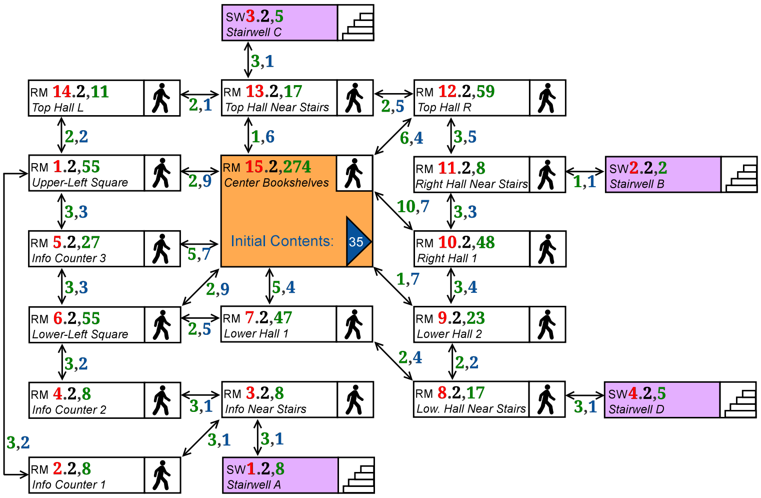


Fig. 4. EvacNET model of the library, second floor.

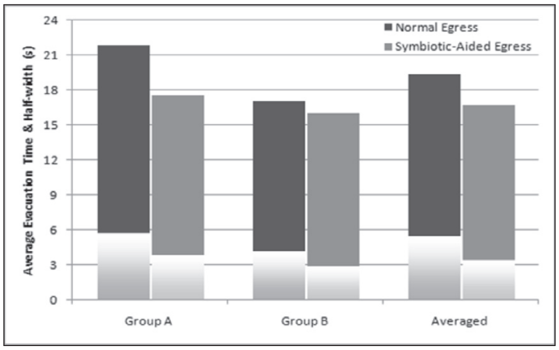
4 depict the full graphical model of the university library using the standard EvacNET syntax[2] from [58]. Two hazard sets were introduced to restrict flow in the building and thus hinder egress. The first was intended to only slightly reduce the flow, while the second would severely impede movement. Both hazard sets only affected a small part of the model, thus allowing informed users to skip the congestion entirely. The "What-If?" simulation used was deterministic in nature. This did not affect the search of such a small physical space; however, if we were to study a larger egress area we would consider switching to a stochastic model as recommended in Section 3.

All experiments excluding the validation exercises took place in the virtual library world. A total of 38 participants took part in this study. Users were given time to explore the virtual building before the start of the simulation, and they performed a mock egress with no hazards in place. Both exits and all four staircases were made known to users during this phase, and a floor plan of the library was displayed in a central location. Hazard sets were not disclosed during the exploratory phase. Prior to the actual experiments, a set of validation runs were recorded, which matched user movement through simple room geometries to the real-world recordings of [56]. Excluding the validation runs, there were four comparative and four progressive simulation runs. Participants in the comparative runs were split evenly into two groups, one of which received symbiotic guidance on the first run, the other on the second. This was intended to counter the natural learning effect from one run to the next. The comparative runs used hazard set 1, and were designed to objectively measure the benefit of symbiotic feedback to egress. We expected symbiotic simulation to offer a definite advantage. The progressive runs used hazard set 2, and required all users to be online at the same time. The purpose of these runs was to see if symbiotic simulation had any noticeable long-term effects on egress time. Our hypothesis was that learning and a general increase in confidence would lead to reduced egress time overall.

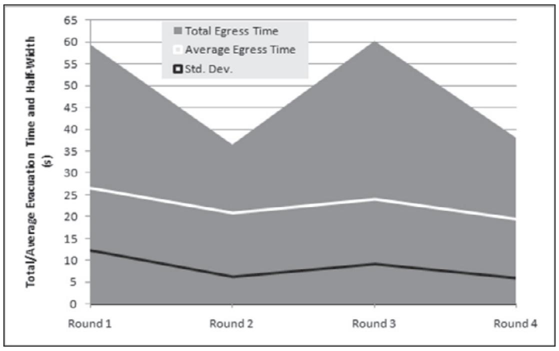
All participants' computers were connected to the same local network, with ≤ 3 ms round-trip network latency between each computer and the server. Two neighbouring computer labs were used to hold all the participants and the server; combined, the experiments took four hours to complete.

In the interest of examining the scalability of perennial simulation, an additional set of experiments was carried out in the virtual library environments. We were primarily concerned with determining the per-agent overhead of experimental

[2] We have extended the official EvacNET specification with the following syntax: **RM1.2↔RM14.2** represents a bi-directional link between **RM1.2** and **RM14.2**, and hallways were represented as pedestrian walking icons rather than face icons.



(a) Average evacuation time, with and without symbiotic simulation.



(b) Evacuation time ordered sequentially, exhibiting some evidence that symbiotic simulation enhances learning.

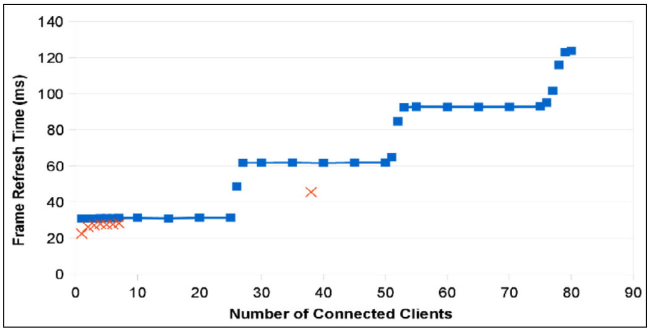
Fig. 5. Evacuationtime analysis.

MMOHILS. In addition, we attempted to measure the space-wise overhead of our perennial simulation, as well as the cost of symbiotic exploration via “What-If?” analysis. For practical reasons, agent scalability was tested using **headless** randomly walking game clients which performed no graphical processing, thus allowing a large number to be multiplexed onto a single machine. An Intel Core i7 MacBook Pro (8 effective cores) with Java 1.6, 64-bit server edition was used to host both the headless clients and the server. Our second performance measurement concerned the scalability of the size of our virtual world. A population of 50 headless clients was randomly scattered throughout an area ranging in size from 0.25 to 72.00 km^2.

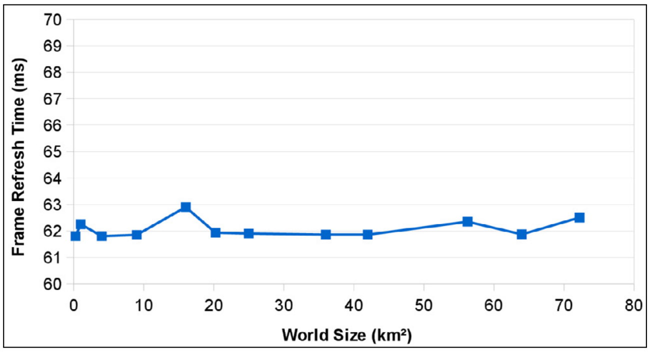
4.1.2. Observations and results

The building egress study featured 25% shorter evacuation times with the introduction of symbiotic feedback. The EvacNET model suggested routes that avoided potential bottlenecks, and real-time goal-directed instructions helped guide occupants out of the building quickly and efficiently. Fig. 5(a) shows the average evacuation time; for groups A and B, and then the two combined. The left and right bars represent the normal and symbiotic egress times, respectively, with the white segments denoting one standard deviation. Improvements of 14% with a 37% reduction in standard deviation were observed for the symbiotic egress scenarios. It can be seen that the symbiotic runs were more cohesive than the normal runs, in addition to being faster. This implies that symbiotic simulation helped to bring individual evacuation time closer to some optimum value.

Fig. 5(b) shows the egress times of all users after several iterative runs of the second hazard set. Rounds 1 and 3 use no symbiotic guidance, while rounds 2 and 4 feature symbiotic simulations. Total egress time is shown in the background; the white and black lines represent average egress time and one standard deviation of the average time respectively. Although total egress time remains roughly unchanged between pairs of like rounds, the average time to evacuate the building clearly decreased after repeated runs. Normal egress times dropped from 26.6 $\pm$ 12.3s to 24.1 $\pm$ 9.2 s, and symbiotic egress times dropped from 20.9 $\pm$ 6.3 to 19.5 $\pm$ 6.0 s. Both normal and symbiotic egress improved by approximately the same amount; however, the standard deviation was reduced by 25% in the latter. Users were already familiar with the building



(a) MMOHILS as client connections increased. The upper data points (blue squares) were measured using headless clients, while the lower data points were gathered on physical, networked machines.



(b) MMOHILS client with 50 connected users as world size increases.

Fig. 6. Performance of MMOHILS.

environment, so the reduction in variance was mostly due to the impact of symbiotic-guided egress. We expect that users had more confidence evacuating the building when presented with authoritative advice from the symbiotic simulator. This is consistent with other research in the field, such as [59] and [60].

The scalability results are presented in Fig. 6(a). For the sake of discussion, a frame refresh of 90 ms corresponds to about 11 fps, which is approximately the limit for acceptable real-time virtual experiences [61]. The virtual (upper) data points were obtained with headless clients, while the real (lower) data points were measured in our previous work [62]. Our actual experiments in [49] noted that, when all 38 test users were connected, latency was 45.5 ms. This is consistent with the “stepping” pattern that occurs as demand on the server increases. The cause of this stepping pattern is well-known to video game developers: when a frame update deadline is missed, the current update is delayed until the next frame, leading to a net loss (in this case) of 30ms regardless of the actual increase in latency.

Several aspects of the scalability results merit further discussion. First, although the machine used for testing virtual demand was much more powerful than the machines used to measure real demand, there is about 10 to 15 ms of overhead on the virtual data points. We expect that the difference in performance is either due to overhead imposed by task switching at the operating system level, or due to performance differences of JVM on Windows versus OS X. Second, we observe that the plateaus in Fig. 6(a) are evenly-spaced: the first occurs after 25 connected users, the second after 50, and the third after 75. When a borderline number of agents was connected, updates would only occasionally miss the frame boundary, leading to values outside the plateaus. Overall performance was not affected by Java’s threading model: the 60-user study was rerun with half of the agents forced onto a separate JVM (and thus a separate core), and performance was observed to be the same. As a result, we can reasonably expect that using more powerful servers would increase the number of agents which can be simulated at each plateau, leading to a cumulative boost in performance and allowing more simultaneous participants in the MMOHILS. Finally, the memory consumption of the simulation and its components did not increase as world size increased, as can be seen from Fig. 6(b).

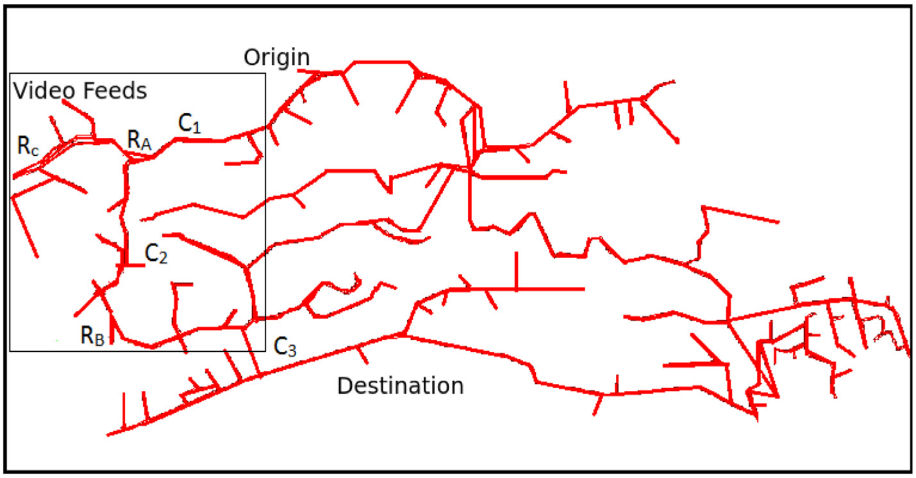


Fig. 7. Screenshot of the enlarged area network running in the MITSIM X11 visualizer, rotated by 45 deg. Various key roads have been labelled.

Although we coined the term MMOHILS from the MMOG community, as mentioned earlier in the introduction, the term *massive* here refers to numbers of the order of magnitude of 10 s to 100 s. We wanted to test the proposed framework for at least 100–1000 of participants. The major challenge faced in conducting such an experiment with such a large number of users is how to incentivize students/people to participate in the study. A review of existing research and development efforts in large-scale virtual worlds is presented in [50], and indicates that world size is unlikely to be a limiting factor in large scale virtual world simulations.

4.2. Incident response (Traffic) scenario using What-if analysis

The second study applied a traffic simulator to investigate incident response on a university campus in Singapore [63]. This focused on providing symbiotic decision support based on vast amounts of real-time data coupled with real-time image processing. The motivation was to optimize dispatch of security personnel based on estimated current traffic conditions and predicted future traffic patterns obtained via simulation. When a situation arises, implementers can run several “What-If?” simulations to determine how to respond optimally: by vehicle or by foot.

4.2.1. Experimental setup

Fig. 7depicts the road network used for this study. The road network was extracted primarily from the Open Street Map data set [64], with a small amount of manual clean-up. Since the simulators used expect traffic to flow on the right (and in Singapore the reverse is true), the road network was flipped about the Y-axis before being sent to the simulation engine. Only one intersection had a traffic signal; its timing was extracted from camera footage at that location. Security camera feeds were available for the area enclosed in a dotted-lined box in Fig. 7. Traffic footage for one week, recorded daily from 3pm to 4pm, was processed to retrieve the arrival time, estimated velocity, and on-screen entrance of each vehicle that passed within range of each camera. Our approach relied heavily on the ViBe background elimination algorithm [65] and Perreaults median filter [66], along with a customised segmentation algorithm described in [63]. ViBe is notable for its performance, resistance to camera jitter, and modest requirement of only a single frame of startup data. Perreaults approach uses an array of past kernel values to compute each pixels filtered value in constant time with respect to kernel size. The image processor was implemented as a series of GStreamer plugins. GStreamer is a multimedia processing framework built around the idea of a software pipeline. Audio and video streams flow from sources through a series of mutative nodes until they reach a sink and terminate. The complete pipeline, shown in Fig. 8, is broken into four phases and two component groups. The resultant image processing algorithm performed well on real-world camera feeds, exhibiting poor results only when multiple sources of noise were present simultaneously.

The simulation itself consisted of the physical world being studied and a virtual world that was used for evaluating the different “What-If?” scenarios in parallel. These simulations were performed using the Simulation of Urban Mobility (SUMO) package [67], with the objective of optimizing the security personnel dispatch by answering the question- “How long would it take for a security vehicle to arrive at the scene of a given incident?” In a real world, such knowledge could be used to avoid dispatching vehicles that will not arrive on time, sending officers by foot instead. For the purpose of evaluation, a secondary simulator (MITSIMLab) was used to confirm whether or not these decisions were actually accurate [68]. MITSIMLab is a microscopic traffic simulation model that allows one to simulate and eva Boththe impact of various Intelligent Transporation System (ITS)strategies at the operational level, and will also be used in the third case study. siluate mulators had their inputs varied in order to induce more variability (stochasticity) into the system. This allowed them to explore the solution space better and react to unexpected events.

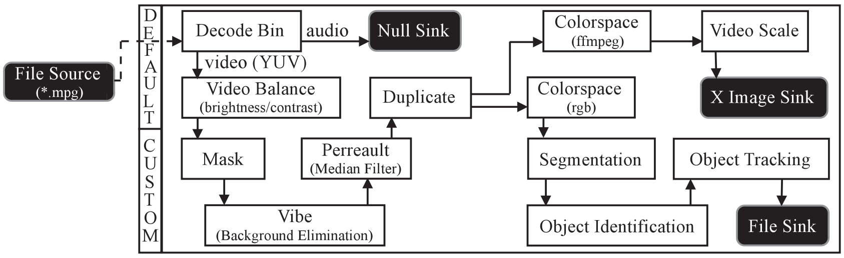


Fig. 8. GStreamer pipeline for our image processing algorithm, with sources and sinks as black rounded rectangles. "Default" plugins are provided by GStreamer; the Custom plugins were developed internally or ported from existing projects.

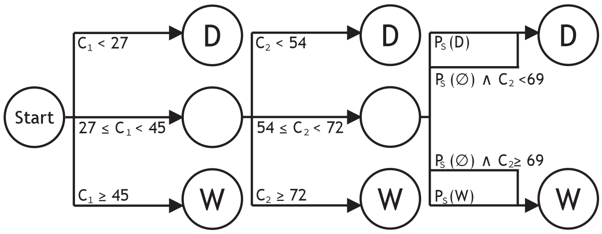


Fig. 9. Decision tree representing incident response strategy on the road network. Nodes marked with a D or W indicate a decision to drive or walk to the location of incident.

Vehicles were grouped into categories with similar properties. Cars, taxis, motorcycles, and trucks each had different driving behaviour characteristics as defined in [69], which were found to be consistent with our observed traffic data. City buses and campus shuttles functioned as heavy vehicles with the additional property of periodically stopping to pick up and drop pedestrian passengers. A vehicle's behaviour was modified by the driver's imperfection parameter; this is a common technique for introducing driver variability into traffic models. The route list was generated each time a "What-If?" simulation was requested by the controller, and was based on real time data extracted from the video feeds. Buses and shuttles were generated first, as they appeared on a regular schedule. For each bus or shuttle arrival, the nearest arrival time from the input data was tagged as being a public transport vehicle, and given a predefined route including bus stops. Then, the remaining entrance arrivals were tagged as one of the remaining vehicle types. Each of these types was given a route based on the observed node choice probability at each intersection.

Three areas were analysed for congestion, marked as R_A, R_B, and R_C in Fig. 7. Region A was centrally located near the main campus roundabout, while Regions B and C were located on the periphery of the campus. To alter congestion levels, the inverse of the mean inter-arrival time was modified by a constant "congestion" factor at each entrance, while route choice probabilities at each node remained unchanged. Twenty experiments were run at each region, with a total of four congestion factors analysed. For each of these experiments, an incident was introduced into the simulator at the same point in time. When an incident entered the system, the controller immediately responded by spawning off a series of "What-If?" simulations. Approximately thirty of these simulations were required to reach an acceptable error threshold.

Following this, a secondary set of experiments was run to answer the question of how perennial simulation compares to traditional simulation-guided preparedness. The entire network was used, with traffic flow estimated from the existing camera data. For these experiments, the goal was to quantify how often the decision to dispatch was correct when using perennial simulation versus traditional simulation.

Evaluating this required us to measure the effect of new information on Implementers at the time of an incident. From existing research, we know that Implementers would likely maintain their decision-making capabilities even through stressful situations [70]. In addition, we feel they are likely to respond positively to simulation-generated advice, based on our own research into symbiotic decision support as well as existing research such as [71] and [72]. Based on this, we chose to model the Implementers' role in this comparison study using decision trees. A decision tree is a directed, acyclic graph used to formalize concepts in decision-support analysis. Decision trees represent decision points and the criteria for moving between them using traditional graph vertices and edges. Fig. 9 depicts the decision tree used to represent the Implementers' decision-making process. At the time of incident, an Implementer must choose whether to dispatch security personnel by foot or by vehicle. The current congestion levels for two major roadways along the shortest driving route are known to Implementers. These are expressed as vehicle density values C_1 and C_2, and are sufficient for making an informed decision.

As shown in Fig. 9, the only combination of values with uncertainty is a C_1 between 27 and 45, coupled with a C_2 between 54 and 72. Only 8.27% of our training data met this criteria; thus, the decision trees represented a typical expert user with reasonable limitations. For such cases, several "What-If?" scenarios are run, and are classified according to the summed squared difference between C_1 and C_2 in the current situation versus the "What-If?" analysis. The resultant travel times of the various "What-If?" scenarios are weighted by this sum, and are further scaled by a small amount to represent

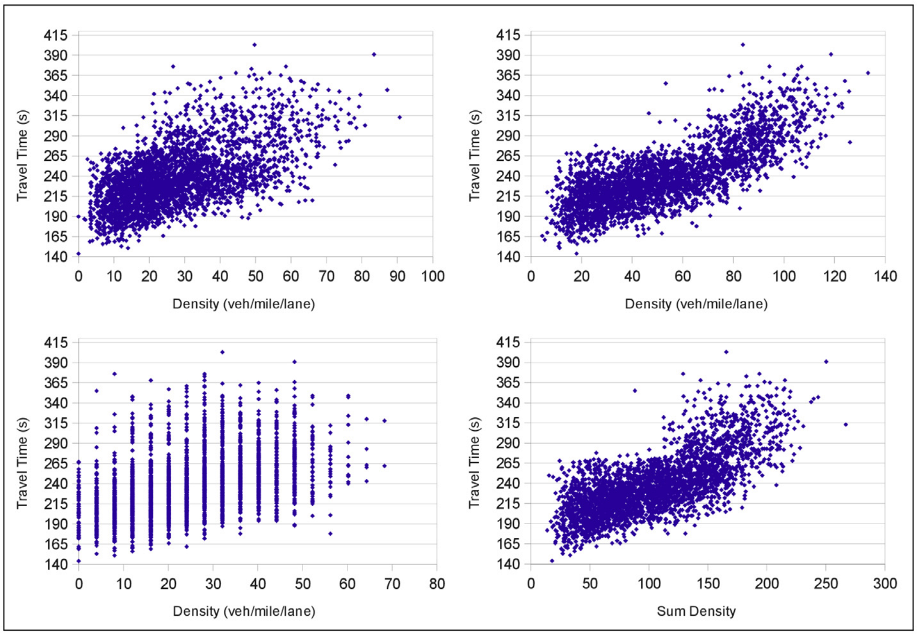


Fig. 10. Data used to train the decision tree. From left to right, top to bottom: congestion parameters C1, C2, and C3, as well as the sum total of all three parameters.

human tendency towards optimistic and emotional bias. Scenarios which predict success are weighted higher, while failure predictors receive less weight. The scenarios are then averaged to provide a general recommendation: either dispatch drivers ($P_S(D)$) or walkers ($P_S(W)$). The Implementers will follow this advice, unless it is unavailable ($P_S(\phi)$), in which case they will use the best estimate from historical data, and dispatch a driver if C_2 is less than 69 or a walker otherwise.

The constants in Fig. 9 were estimated from a set of 3000 vehicle traces at various congestion factors ranging from 1.0 (low congestion) to 3.8 (high congestion). For each vehicle trace, the values C_1, C_2, C_3 and the total trip time in seconds were known. Although undesirable, false positives are considered less damaging than true negatives, as the latter leads to a situation where no security personnel reach the area of incident in time. As such, false positives received half the weight of true negatives when assessing the severity of an incorrect incident response. Fig. 10 contains a plot of trip time for each congestion parameter, as well as for the combined sum of congestion, for reference. C_1 is a strong indicator of trip time, and C_2 is nearly as accurate. C_3 suffers from artifacts typical to MITSIM for short, straight road segments, leading to a striated pattern. In addition, its influence is comparatively weak, most likely due to its location far downstream from the security vehicle's origin. Further analysis of the data led us to conclude that C_1 provides the most stable indicator of trip arrival time, with C_2 covering the areas where C_1 is not consistent. For some combinations of C_1 and C_2, it was impossible to predict success or failure with an accuracy of 10% better than chance. These situations were expected to benefit the most from perennial simulation. In addition, although the data used to estimate the parameters in the Implementers' decision tree is assumed to be free from of error, some amount of uncertainty was added to the actual data at the time of each incident. An estimate of this error value is also made known to the Implementers and, in the case of symbiotic guided feedback, incorporated into the "What-If?" simulations. Regardless of how the Implementers arrived at their decision, that decision will then be evaluated by the transit simulator.

4.2.2. Observations and results

The incident response study's results are shown in Fig. 11. The "What-If?" experiments were combined to form a "response radius" for each scenario. Any vehicle which could reach the area of incident within a given time limit was considered within the response radius. Region A's response radius decreased slightly as congestion increased, but remained relatively unaffected overall. This is likely due to its central location near the intersection of three major roadways. Regions B and C each noticed a steady drop in response radius as congestion increased, reflecting the remoteness of these locations and their subsequent vulnerability to increased congestion. Region B's response radius plummeted under a congestion factor of 2.0, which is particularly noteworthy as 2.0 only represents a medium-high level of congestion. Recall that the default congestion factor of 1.0 was extracted from non-peak traffic patterns from 3 pm to 4 pm. The data gathered expose several non-obvious facts. For example, although Region A is obviously easier to reach than the others, discerning a difference between the equally-remote Regions B and C is only possible through simulation. The difference in their response radii, and Region B's sensitivity to congestion are both valuable insights that can aid dispatchers at the time of incident. As the

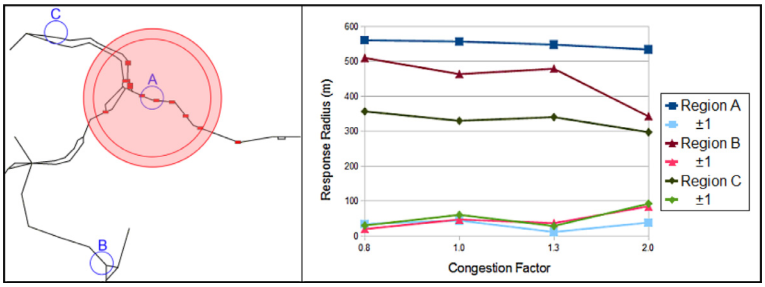
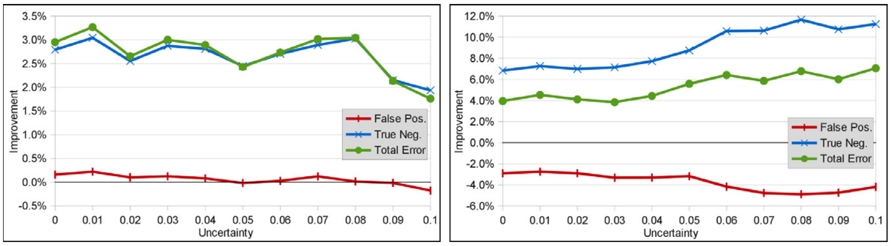


Fig. 11. Left: Campus road network, with three regions marked A, B, and C. Region A incident response radius and variance are depicted for congestion factor 1.0. Right: Mean response radius (1 standard deviation) for each region under different congestion factors (scaled O/D arrival rates).



(a) Control scenarios. (b) Testing scenarios.

Fig. 12. Improvement offered by perennial simulation.

complexity of the road network increases, the ability to back up "obvious" statements with actual experimental evidence becomes essential.

Regarding the image processing algorithm, performance was acceptable for real-time operation, and all of the image processing pipeline shortcomings have straightforward workarounds which are discussed in [63]. Thus, it is reasonable to assume that it could run unattended in real time.

Regarding the comparison of perennial to traditional simulation, Fig. 12(a) and (b) depict our results for the Control and Testing scenarios, respectively. The former contained points that were clearly defined in the decision tree, and represented decisions that were not considered likely to benefit from perennial simulation. The latter contained points the decision tree was unable to provide advice on, and were considered primary candidates for improvement. Each graph contains 47,000 simulation traces of dispatched vehicles for several uncertainty values from 0.0 to 0.1. The difference in predictive performance is represented in terms of true negatives and false positives, as well as a combined improvement in total errors.

The Control group featured general overall improvement and better true negative avoidance, both of which hovered around 1.8 to 3.2%. False positive avoidance generally remained positive, and had no real impact on the overall performance of the system. These results are typical of what one would expect with a control group -the use of perennial simulation in cases where it is not strictly needed does not offer a significant improvement over traditional simulation methods. The Testing group was necessarily more difficult to reason about, even if information is perfect. In this case, perennial simulation incurred a slight penalty in terms of false positives, but countered this with a large improvement in true negative reduction. The overall trend line is positive, and increases from 4.0% to 7.2% as uncertainty increases. In other words, the perennial simulation is more than twice as good at reducing error rates as it was in the Control group, and this predictive power is robust in the face of data uncertainty. It compensates by dispatching more vehicles than are strictly necessary, but as stated before, true negatives are sufficiently more damaging than false positive to justify this, and the overall trend line remains positive.

4.3. Guidance based traffic management during crisis

For the third case study, a closed loop approach is deployed which integrates a microscopic traffic simulator MITSIMLab, a mesoscopic traffic simulator DynaMIT along with the GStreamer pipeline for data acquisition and data processing from different sources like traffic cameras and sensor feeds. With this closed loop approach, the *recurring* (symbiotic interaction between MITSIMLab and DynaMIT) and *enduring* (different "what-if?" simulations in DynaMIT) attributes of the perennial simulation is exploited. Some descriptions on MITSIMLab and DynaMIT are necessary before we describe the case study.

4.3.1. MITSIMLab

MITSIMLab (microscopic traffic simulation laboratory) is a microscopic traffic simulation model that evaluates the impacts of alternative traffic management system designs, travel information systems, public transport operations, and various

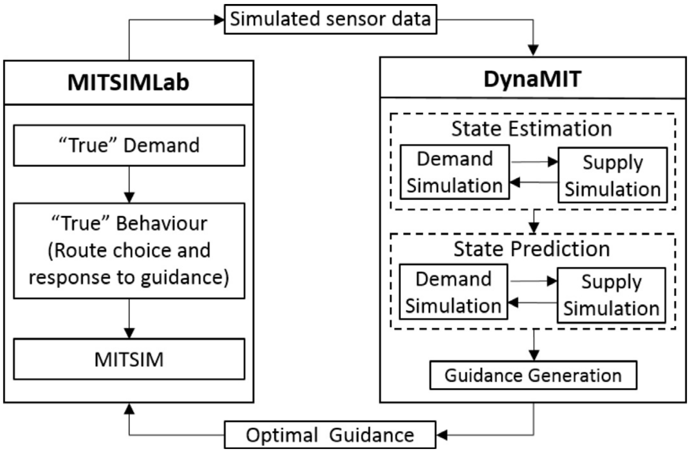


Fig. 13. The closed loop setup between MITSIMLab and DynaMIT, forming a symbiotic feedback to each other.

ITS strategies at the operational level and assists in their subsequent refinement. MITSIMLab can evaluate systems such as advanced traffic management systems (ATMS) and route guidance systems. The core of MITSIMLab consists of travel and driving behaviour models. The travel behaviour models capture the drivers pre-trip and en route choices. The driving behaviour models deal with tactical and operational driving decisions, mainly acceleration and lane changing. The models that capture these choices in MITSIMLab are probabilistic, based on the theories of random utility maximization. A full description along with experiments on different real world networks can be found in [73].

4.3.2. DynaMIT

Dynamic Network Assignment for the Management of Information to Travellers (DynaMIT) [74] is a simulation-based DTA model system that estimates and predicts traffic conditions. Internally DynaMIT consists of two simulators: demand and supply. The demand simulator models the detailed travel demand, behaviour and their complex interactions. The mesoscopic supply simulator captures traffic dynamics and evaluates the performance of the network using speed-density relationships and queuing theory. The complex demand-supply interactions are represented by algorithms that integrate real-time data obtained from traffic surveillance system with other information accessible in historical database to estimate the current network state, predict future conditions, and help in generating anticipatory route guidance and control strategies. Because of its real time traffic prediction capability DynaMIT is used in real-time decision making using different network control strategies.

4.3.3. Experimental setup

For the case study, MITSIMLab simulates the real world traffic conditions, estimating the current state of the traffic condition from the information provided which can be real time or historical. Using the output from MITSIMLab, DynaMIT will itself estimate the traffic conditions. Based on the estimated conditions, DynaMIT will subsequently simulate and predict the future traffic conditions. It is to be noted that DynaMIT does not receive any real world data, it only receives data from MITSIMLab. Therefore, MITSIMLab acts as a proxy or surrogate real world and provides the necessary real time data to DynaMIT. Based on the predicted traffic conditions, DynaMIT will provide optimal guidance and strategy to MITSIMLab. Guidance, in the form of link travel times and strategy in the form of tolls/vms/ramp-metering comprise the closed loop symbiotic feedback to maintain consistency. Optimal strategy is generated using "What-if" analysis. Fig. 13 depicts the approach more clearly. A detailed discussion and implementation of this closed loop approach can be found in [75].

The main objective is to have a good estimate of the future traffic conditions (i.e. DynaMIT State Prediction Fig. 13) to disseminate optimal guidance to travellers on the road network. In order to do prediction, the existing network conditions, i.e. real time data is required, which in this case is provided by MITSIMLab. This type of framework makes the system robust enough to deal with situations when the real data available is noisy, sparse or not reliable, which is generally the case during crisis. The above outlined approach is validated on a synthetic test network consisting of 14 nodes, 19 links, 43 segments and 12 OD pairs as shown in Fig. 14. 11 sensors are used to collect surveillance data, mainly speed and counts. To maintain consistency and make the whole setup more close to the real world scenario, DynaMIT is calibrated based on the output of the MITSIMLab. The calibration mainly involves OD pairs, route choice, segment capacities and speed-density relationship parameters. The calibration problem is formulated as an optimization problem [76].

4.3.4. Observations and results

Three scenarios are considered to test the efficacy of the generated strategies using this approach: (1) Base case without incident and route guidance, (2) Incident but no route guidance and (3) Incident with route guidance. Scenario (1) acts as a base case when the network operates normally without any incident. The effect of route guidance is measured using the

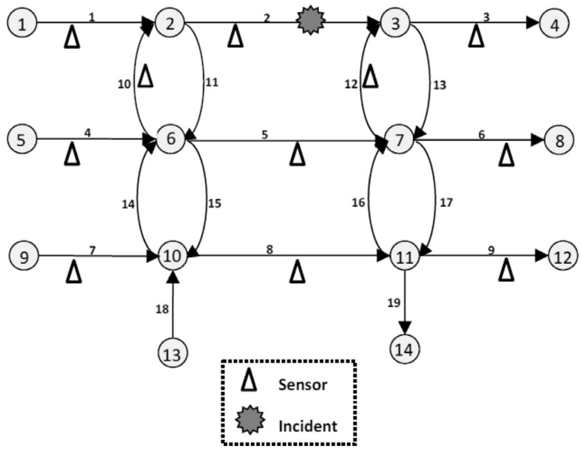


Fig. 14. Synthetic test network.

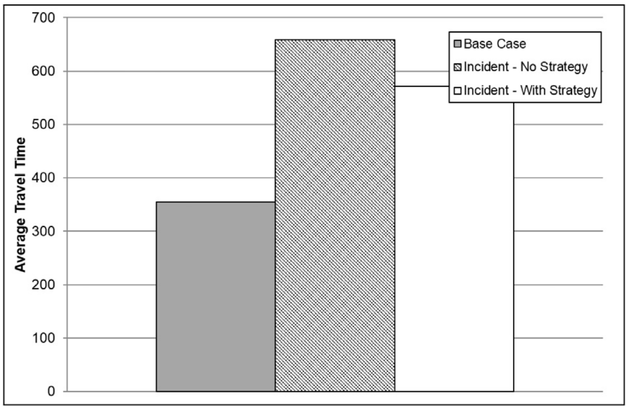


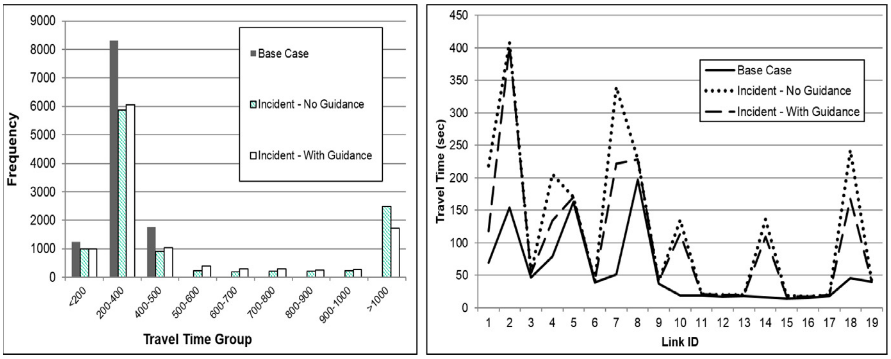
Fig. 15. Total average travel time for three scenarios.

difference in traffic conditions of scenarios (2) and (3). To have a crisis-like situation, a severe incident is modelled with segment capacity reduced up to 20% of its original value. The incident is located on link 2 (Fig. 14) from 7:30–8:30 am.

To evaluate the effectiveness of the route guidance strategies generated by DynaMIT, travel time for each vehicle finishing the trip is collected from MITSIMLab output. As MITSIMLab acts as a surrogate for the real world, all performance metrics are calculated from the output of MITSIMLab. Fig. 15 shows the average travel time for all the three scenarios, it is clearly evident that in the case of incident the network conditions improve if strategy is implemented. In the base case, the average travel time for all OD pairs is 354.6 s. When incident happens and no route guidance is provided, the travel time drastically increases by 85.6% to 658.3 s. With route guidance, the average travel time increases by only 61.1% to 571.3 s. This implies that 28.6% of the increase in average travel time is eliminated by the route guidance strategy.

Fig. 16(a) plots the frequency distribution of travel times for the three scenarios. In the base case, all the drivers belong to the first four groups with low travel times. When incident happens and no route guidance is provided, a large number of drivers shift to the last group with very high travel time. When route guidance is provided, the number of drivers with very high travel time decreases. This is a meaningful effect since it improves the satisfaction of the travellers in the network. The average link travel times for the three scenarios are shown in Fig. 16(b). The solid line shows the travel time for the base case, the dashed line shows travel times when there is incident. The dotted line shows the travel times when incident is present but guidance is provided. The considerable reduction in the link travel time is due to the guidance strategy which mitigates congestion by diverting drivers from the congested link to other links with high residual capacity.

DynaMIT takes about 11 s to perform a single “what-if?” simulation, this includes loading the sensor data (from MITSIMLab) , state estimation and prediction. To achieve a real-time performance, multiple state predictions (in DynaMIT) can be simulated in parallel on a multi-core architecture. This becomes more pertinent in crisis scenarios where guidance or strategy should be provided in real time.



(a) Frequency of Travel Times. (b) Average link travel times for three scenarios

Fig. 16. Travel time analysis.

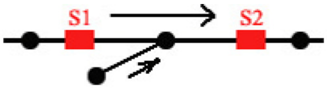


Fig. 17. On-ramp traffic segment.

4.4. General discussion

The results of the first two case studies presented require some additional context in the form of real-world considerations. First, the results obtained in the building egress study relied on accurate and ubiquitous sensors, but real-world deployments would likely encounter noise and uncertainty. In particular, sensing the total number of users in real-time would be complicated by the fact that building occupants might not have a tracking device (for example their phones turned on). We expect that as the inaccuracy of the sensors and effectors increases, the total benefit gained from symbiotic simulation would decrease. On the other hand, the virtual library environment was both very simple and sparsely populated. Truly disastrous egress conditions tend to arise when neither of these conditions holds, so it is fair to expect that most real world environments would benefit enormously from symbiotic feedback.

The results of the second case study, provides some justification for our treatment of noise and uncertainty. In particular, it is promising to find that perennial simulation improves over the control group. Although, a 3% improvement is not quite significant for setting up perennial simulation, but if the system is already present for the testing cases, then Implementers may take advantage of the system in mundane cases as well. In addition, as noise increased, the perennial simulation outperformed traditional information gathering methods (until both collapsed at uncertainty values higher than 0.1).

To deal with noisy sensor data or faulty sensors, one approach which can be adopted and one that we have tried is sensor data cleansing. Traffic sensor data available may be "noisy" and inconsistent. To overcome this, we use historical sensor counts together with a heuristic algorithm to clean (remove) the noisy sensors and perform offline calibration to test the efficacy of our cleansing algorithm. For example, if all of a sudden, the sensor count falls to zero, when historically, it is non-zero, then we should be concerned with that sensor. We also perform testing to see if the sensor data makes sense, e.g. if we have a road segment with an On-ramp (Fig. 17), then sensor counts at the downstream (s2) should be higher than the upstream counts (s1). Based on these tests, we decide which sensors are faulty and discard their data from the system. These algorithms are included in the converter/agglomerator in the sensescape part of the framework.

The third case study mainly deals with traffic management under crisis, and validates the the efficacy of guidance and strategy in traffic management. It also provides another approach (apart from data cleansing) to deal with situations where data may not be available, but a surrogate microscopic simulator which has been extensively calibrated can be a promising alternative. A more comprehensive approach using a real world road network and real time data would certainly provide a better validation.

5. Conclusions and future work

We presented a perennial simulation framework that targets crisis management simulation. This framework incorporated concepts of dynamic data-driven systems, symbiotic simulation as well as human-in-the-loop techniques. We detailed the framework and its components, and described three experiments that used the framework and served as case studies.

The first case study described a building evacuation scenario, and demonstrated that symbiotic simulation can improve the egress times of building occupants if real time guidance or information is provided. The second case study dealt with incident response on crowded roadways, and demonstrated the real time decision making capability of perennial simulation.

The third case study showed a novel approach of coupling two simulators, where one simulator was used for providing guidance, while the other acted as a surrogate model for the real world. Taken together the case studies provide a strong motivation for setting up perennial simulations: they are long-lasting, provide benefits during non-crisis situations, and are pivotal in establishing information superiority during a crisis an event in which time is a scarce resource, and any informed decision or control is welcome.

As future work, we aim to establish an upper bound on the runtime of "What-If?" simulations and validate case study 3 on Singapore's road network. We aim to speed up the performance of "What-If?" simulations by using parallel computing techniques like multi-core architectures or GPUs.

Acknowledgments

This work is supported by Ministry of Education, Singapore Academic Research Fund R-252-000-314-112 and the National Research Foundation Singapore through the Singapore-MIT Alliance for Research and Technology's Future Urban Mobility grant.

References

[1] E. Larson, Isaac's Storm: A Man, a Time, and the Deadliest Hurricane in History, First, Knopf Doubleday Publishing Group, 2000.
[2] N.L. Frank, The great Galveston hurricane of 1900, in: Hurricane! Coping with Disaster: Progress and Challenges Since Galveston, 1900, American Geophysical Union, 2013, pp. 128–140.
[3] G. Giuliano, K. Haboian, K. Rutherford, J. Prashker, W. Recker, Evaluation of 1984 Los Angeles Summer Olympics Traffic Management, Technical Report, Institute of Transportation Studies, University of California, 1987.
[4] E. Rowe, The Los Angeles automated traffic surveillance and control (ATSAC) system, IEEE Trans. Veh. Technol. 40 (1) (1991) 16–20.
[5] C. Smith, S. Hayne, A distributed system for crisis management, in: Proceedings of the 24th Annual Hawaii International Conference on System Sciences, 3, 1991, pp. 72–81.
[6] T. Bui, R. Subba, A tale of two disasters: assessing crisis management readiness, in: Proceddings of the 42nd Hawaii International Conference on System Sciences, 2009, pp. 1–10.
[7] R. Desimone, The application of ai planning technology to crisis management tasks, in: IEE Colloquium on Intelligent Planning and Scheduling Solutions (Digest No. 1996/197), 1996, pp. 6/1–6/3.
[8] S. Tufekci, W. Wallace, The emerging area of emergency management and engineering, IEEE Trans. Eng. Manag. 45 (2) (1998) 103–105.
[9] I. Martinez-Moyano, S. Conrad, E. Rich, D. Andersen, Modeling the emergence of insider threat vulnerabilities, in: Proceedings of the 2006 Winter Simulation Conference (WSC'06), 2006, pp. 562–568.
[10] U. Gupta, N. Ranganathan, Multievent crisis management using noncooperative multistep games, IEEE Trans. Comput. 56 (5) (2007) 577–589.
[11] J. Lee, T. Bui, A template-based methodology for disaster management information systems, in: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2, 2000, pp. 1–7.
[12] B. Belkhouche, R. Bakeer, S. Al-saleh, An Expert Decision Support System for Crisis Management, EECS Department Tulane University, New Orleans, LA 70118, 2002.
[13] M.A. Finney, FARSITE: Fire area simulator-model development and evaluation - Technical Report RMRS-RP-4, Technical Report, United States Department of Agriculture - Forest Service, 2004.
[14] C. Douglas, R. Loader, J. Beezley, J. Mandel, R. Ewing, Y. Efendiev, G. Qin, M. Iskandarani, J. Coen, A. Vodacek, M. Kritz, G. Haase, Dddas approaches to wildland fire modeling and contaminant tracking, in: Proceedings of the 2006 Winter Simulation Conference (WSC'06), 2006, pp. 2117–2124.
[15] D. Vakalis, H. Sarimveis, C.T. Kiranoudis, A. Alexandridis, G. Bafas, A gis based operational system for wildland fire crisis management ii. system architecture and case studies, Appl. Math. Modell. 28 (2004) 411–425.
[16] L. Hernández Encinas, S. Hoya White, A. Martín del Rey, G. Rodríguez Sánchez, Modelling forest fire spread using hexagonal cellular automata, Appl. Math. Modell. 316 (6) (2007) 1213–1227.
[17] T. Ghisu, B. Arca, G. Pellizzaro, P. Duce, An improved cellular automata for wildfire spread, Procedia Comput. Sci. 51 (2015) 2287–2296.
[18] T. Lant, M. Jehn, C. Christensen, O.M. Araz, J.W. Fowler, Simulating pandemic influenza preparedness plans for a public university: a hierarchical system dynamics approach, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 1305–1313.
[19] A. Ekici, P. Keskinocak, J. Swann, Pandemic influenza response, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 1592–1600.
[20] M. Brandeau, Infectious disease control policy: A role for simulation, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 1578–1582.
[21] D.J. Medeiros, E. Swenson, C. DeFlitch, Improving patient flow in a hospital emergency department, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 1526–1531.
[22] R.E. Giachetti, A simulation study of interventions to reduce appointment lead-time and patient no-show rate, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 1463–1468.
[23] F. Fiedrich, An hla-based multiagent system for optimized resource allocation after strong earthquakes, in: Proceedings of the 2006 Winter Simulation Conference (WSC'06), 2006, pp. 486–492.
[24] A. Shendarkar, K. Vasudevan, S. Lee, Y.-J. Son, Crowd simulation for emergency response using BDI agent based on virtual reality, in: Proceedings of the 2006 Winter Simulation Conference (WSC'06), 2006, pp. 545–553.
[25] W.H.O.G.I. Programme, Pandemic influenza preparedness and response: a WHO guidance document, World Health Organization, 2009.
[26] M. Low, W. Cai, S. Zhou, A federated agent-based crowd simulation architecture, in: The 2007 European Conference on Modelling and Simulation, 2007, pp. 188–194.
[27] S. Sakellariou, S. Tampekis, F. Samara, A. Sfougaris, O. Christopoulou, Review of state-of-the-art decision support systems (dsss) for prevention and suppression of forest fires, J. Forestry Res. 28 (6) (2017) 1107–1117.
[28] M. Avvenuti, S. Cresci, A. Marchetti, C. Meletti, M. Tesconi, Ears (earthquake alert and report system): a real time decision support system for earthquake crisis management, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 1749–1758.
[29] C. Bielski, V. OBrien, C. Whitmore, K. Ylinen, I. Juga, P. Nurmi, J. Kilpinen, I. Porras, J.M. Sole, P. Gamez, M. Navarro, A. Alikadic, A. Gobbi, C. Furlanello, G. Zeug, M. Weirather, J. Martinez, R. Yuste, S. Castro, V. Moreno, T. Velin, Coupling early warning services, crowdsourcing, and modelling for improved decision support and wildfire emergency management, in: Proceedings of 2017 IEEE International Conference on Big Data (BIGDATA), 2017.
[30] T. Martagan, B. Eksioglu, S. Eksioglu, A. Greenwood, A simulation model of port operations during crisis conditions, in: Proceedings of the 2009 Winter Simulation Conference (WSC'09), 2009, pp. 2832–2843.
[31] T.B. Quillinan, F. Brazier, H. Aldewereld, F. Dignum, V. Dignum, L. Penserini, N. Wijngaards, Developing agent-based organizational models for crisis management, in: Proc. of the 8th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'09), ACM, 2009, pp. 45–51.

[32] P. Dihé, R. Denzer, M. Polese, A.-M. Heikkilä, D. Havlik, J. Sautter, T. Hell, S. Schlobinski, G. Zuccaro, W. Engelbach, An architecture for integrated crisis management simulation, in: Proceedings of the 20th International Congress on Modelling and Simulation, 2013, pp. 1–6.
[33] F. Rousseaux, K. Lhoste, Process analysis, modeling and simulation for crisis management, in: International Workshop on Advanced Information Systems for Enterprises, 2008 (IWAISE'08), IEEE, 2008, pp. 41–45.
[34] R.A. Gonzalez, Crisis response simulation combining discrete-event and agent-based modeling, in: Proceedings of the 6th International Conference on Information Systems for Crisis Response and Management, (ISCRAM'09), 2009.
[35] R. Fujimoto, D. Lunceford, E. Page, Amu: Grand challenges for modeling and simulation: Dagstuhl report, Technical Report 350, Schloss Dagstuhl. Seminar, 2002.
[36] H. Aydt, S.J. Turner, W. Cai, M.Y.H. Low, P. Lendermann, B.P. Gan, Symbiotic simulation control in semiconductor manufacturing, in: Computational Science (ICCS'08), Springer, 2008, pp. 26–35.
[37] F. Kamrani, R. Ayani, Using on-line simulation for adaptive path planning of uavs, in: Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications, IEEE Computer Society, 2007, pp. 167–174.
[38] H. Aydt, S. Turner, W. Cai, M. Low, P. Lendermann, B.P. Gan, R. Ayani, Preventive what-if analysis in symbiotic simulation, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 750–758.
[39] F. Darema, Grid computing and beyond: The context of dynamic data driven applications systems, Proc. IEEE 93 (3) (2005) 692–697.
[40] M.Y.H. Low, K.W. Lye, P. Lendermann, S.J. Turner, R.T.W. Chim, S.H. Leo, An agent-based approach for managing symbiotic simulation of semiconductor assembly and test operation, in: Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'05), ACM, 2005, pp. 85–92.
[41] D. Evans, A. Heuvelink, D. Nettle, The evolution of optimism: a multi-agent based model of adaptive bias in human judgement, in: Proceedings of the AISB'03 Symposium on Scientific Methods for the Analysis of Agent-Environment Interaction, 2003.
[42] S. Lee, Y.-J. Son, Integrated human decision making model under belief-desire-intention framework for crowd simulation, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 886–894.
[43] A. Chaturvedi, A. Mellema, S. Filatyev, J. Gore, DDDAS for fire and agent evacuation modeling of the Rhode Island nightclub fire, in: International Conference on Computational Science (3), in: Lecture Notes in Computer Science, Springer, 2006, pp. 433–439.
[44] E. Rome, T. Doll, S. Rilling, B. Sojeva, N. Voß, J. Xie, The use of what-if analysis to improve the management of crisis situations, in: R. Setola, V. Rosato, E. Kyriakides, E. Rome (Eds.), Managing the Complexity of Critical Infrastructures. Studies in Systems, Decision and Control, 90, Springer, Cham, 2016, pp. 233–277.
[45] F. Darema, Dynamic data driven applications systems: A new paradigm for application simulations and measurements, in: Computational Science (ICCS'04), Springer, 2004, pp. 662–669.
[46] A.M. Uhrmacher, D. Weyns, Multi-Agent systems: Simulation and applications, CRC press, 2009.
[47] S. Singhal, M. Zyda, Networked Virtual Environments: Design and Implementation, ACM Press/Addison-Wesley Publishing Co., 1999.
[48] S.N. Hetu, G. Tan, MMOHILS: a simpler approach to valid agents in human simulation studies, in: Proceedings of the 2008 Winter Simulation Conference (WSC'08), 2008, pp. 909–913.
[49] S.N. Hetu, G. Tan, Potential benefits of symbiotic simulation to pedestrian evacuation, in: Asia Simulation Conference 2009, Japan Society for Simulation Technology, 2009.
[50] S.N. Hetu, A Perennial Simulation Framework for Integrated Crisis Management Studies, National University of Singapore, 2013 Ph.D. thesis.
[51] S.N. Hetu, G. Tan, The big picture of symbiotic decision support: designing a "What-If" simulation framework for crisis management, in: FISAT: Second International Conference on Advanced Computing and Communications Technologies for High Performance Applications, 2010. Keynote address.
[52] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 2nd, Prentice-Hall, Englewood Cliffs, NJ, 2003.
[53] G.K. Still, Crowd Dynamics, Warwick University, 2000 Ph.D. thesis.
[54] D. Helbing, P. Molnar, Social force model for pedestrian dynamics, Phys. Rev. E 51 (5) (1995) 4282.
[55] X. Pan, C.S. Han, K.H. Law, A multi-agent based simulation framework for the study of human and social behavior in egress analysis, in: Proceedings of the ASCE International Conference on Computing in Civil Engineering, 92, 2005.
[56] W. Daamen, S. Hoogendoorn, Qualitative results from pedestrian laboratory experiments, in: Proceedings of the 2nd International Conference on Pedestrian and Evacuation Dynamics, CMS Press, 2003, pp. 121–132.
[57] J.J. Fruin, Pedestrian Planning and Design, Metropolitan Assoc. Urban Des. Environ. Planners, 1971.
[58] T.M. Kisko, R. Francis, C. Nobel, Evacnet4 users guide, University of Florida (1998).
[59] L. Festinger, A theory of social comparison processes, Human Relations 7 (1954) 117–140.
[60] S. Milgram, L. Bickman, L. Berkowitz, Note on the drawing power of crowds of different size., J. Personality Social Psychol. 13 (2) (1969) 79–82.
[61] N.I. Durlach, A.S. Mavor, Virtual Reality: Scientific and Technological Challenges, Natl. Acad. Press, 1994.
[62] S.N. Hetu, G. Tan, Real-time simulation in Java: A feasibility study, in: System Simulation and Scientific Computing, 2008. ICSC 2008, Asia Simulation Conference, 2008, pp. 396–399.
[63] S.N. Hetu, G. Tan, Perennial simulation of a legacy traffic model: Implementation, considerations, and ramifications, in: Proceedings of the 2011 Winter Simulation Conference (WSC'11), 2011, pp. 1519–1530.
[64] OpenStreetMap.org, Open street map: The free wiki world map, 2010. (http://www.openstreetmap.org)
[65] O. Barnich, M. Droogenbroeck, ViBe: A universal background subtraction algorithm for video sequences, IEEE Trans. Image Process. 20 (6) (2010) 1709–1724.
[66] S. Perreault, P. Hebert, Median filtering in constant time, IEEE Trans. Image Process. 16 (9) (2007) 2389–2394.
[67] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent development and applications of sumo–simulation of urban mobility, Int. J. Adv. Syst. Measurements 5 (3&4) (2012).
[68] Q. Yang, A Microscopic Traffic Simulator for evaluation of dynamic traffic management systems, Transp. Res. Part C 4 (3) (1996) 113–129.
[69] S. Krauß, Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics, University of Cologne, Koln, Germany, 1998 Ph.D. thesis.
[70] J. Shanteau, G. Dino, Environmental stressor effects on creativity and decision making, in: Time Pressure and Stress in Human Judgment and Decision Making, Springer, 1993, pp. 293–308.
[71] N. Pennington, R. Hastie, The story model for juror decision making, in: R. Hastie (Ed.), Inside the juror: The psychology of juror decision making, Cambridge University Press, 1993.
[72] L. Adelman, M.A. Tolcott, T.A. Bresnick, Examining the effect of information order on expert judgment, Org. Behav. Human Decision Processes 56 (3) (1993) 348–369.
[73] M. Ben-Akiva, H.N. Koutsopoulos, T. Toledo, Q. Yang, C.F. Choudhury, C. Antoniou, R. Balakrishna, Traffic simulation with mitsimlab, in: Fundamentals of Traffic Simulation, Springer, 2010, pp. 233–268.
[74] M. Ben-Akiva, H.N. Koutsopoulos, C. Antoniou, R. Balakrishna, Traffic simulation with dynamit, in: Fundamentals of Traffic Simulation, Springer, 2010, pp. 363–398.
[75] V.-A. Vu, G. Park, G. Tan, M. Ben-Akiva, A simulation-based framework for the generation and evaluation of traffic management strategies, in: Proceedings of the 2014 Annual Simulation Symposium, in: ANSS '14, Society for Computer Simulation International, San Diego, CA, USA, 2014, pp. 9:1–9:10.
[76] R. Balakrishna, H.N. Koutsopoulos, M.E. Ben-Akiva, Calibration and validation of dynamic traffic assignment systems, in: Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic Theory, 2005, pp. 407–426.