

Application of Microsimulation Towards Modelling of Behaviours in Complex Environments

Daniel Keep, Rachel Bunder, Ian Piper, Anthony Green

School of Computer Science and Software Engineering
University of Wollongong
ian@uow.edu.au

Abstract

In this paper, we introduce new capabilities to our existing microsimulation framework, Simulacron. These new capabilities add the modelling of behaviours based on motivations and improve our existing non-deterministic movement capacity. We then discuss the application of these new features to a simple, synthetic, proof of concept, scenario involving the transit of people through a corridor and how an induced panic affects their throughput. Finally we describe a more complex scenario, which is currently under development, involving the detonation of an explosive device in a major metropolitan transport hub at peak hour and the analysis of subsequent reaction.

Introduction & Background

Microsimulation is a discrete simulation technique which allows for the modelling of the behaviour of individuals in a complex system (Connor et al. 2000; Merz 1991). It was originally devised for financial and economic modelling (Weinstein 2006; Orcutt 1957), but is generally applicable to a wide range of scenarios.

This paper builds on our previous work (Piper et al. 2009; 2010; Green et al. 2009; 2010b; 2010a; Zhang et al. 2008) in which we describe our microsimulation-based approach to modelling systems involving many individuals and their interactions. Our modular microsimulation framework, Simulacron, allows us to quickly build simulations with varying requirements. As covered in previous publications, it has already been applied to problems involving epidemiology and terrorism. This paper will be making use of the framework itself and an updated dispersion module from previous publications.

We will describe new functionality added to our existing toolset in the form of a motivation module. This module allows us to model human interactions involving differing motivators and their effect on behaviour. A simple, fictional scenario will then be introduced to test and demonstrate this new functionality. The results of our experiments with this scenario will be presented and discussed. Finally, we will introduce a more complex scenario incorporating the interaction of the motivation module with a new goal-based movement module, which we briefly describe in this paper.

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

It is important to realise that the work described here involves the incorporation of new functionality into a generalised microsimulation framework and is not a new, standalone packaging of capabilities for a specific task. This means that the newly introduced functionality can be combined with our existing capabilities permitting, for example, the incorporation of psychological factors into an epidemiological simulation; this significantly enhances our modelling abilities.

This development was motivated by interest both within our group and from external agencies in using our techniques to model scenarios such as transit through a transport hub, emergency response behaviours, evacuation, interdiction and evaluation of security mechanisms. We realised we could not adequately model these with our existing toolset and required a more sophisticated approach to behavioural modelling.

Our microsimulation-based approach to modelling divides the world into entities of two classes: “cells” and “peeps”. Cells are abstract locations which have no intrinsic meaning or properties whilst peeps are abstract things whose only intrinsic property is their location (a cell). Cells and peeps have meaning associated with them via “fields” which are arbitrary chunks of data associated by name. This system allows us to extend the existing abstractions with new meaning without having to modify the simulation framework itself.

The field system supports the use of “states” which allow alternate field value sets to be associated with an entity based on their current state. As an example, a peep might have two different schedules: one for use in the “working” state and one for use in the “weekend” state.

The simulation program is the combination of the simulation framework, called Simulacron, and one or more modules. These modules provide the actual simulation behaviour. To date, we have modules implementing cyclic and one-off scheduling, non-deterministic dispersion, spread of infectious agents and a terrorist behaviour prototype. A user is free to use whatever subset of modules they need; the simulation need only be as complicated as the scenario demands.

Input to the simulation is provided in the form of an XML specification which details all entities in the model. Because of the microsimulation approach, this data set must contain

explicit values for each field for each peep and cell. For any real-world simulation, this leads to extraordinarily large data sets which are practically impossible to manually construct.

To alleviate this problem, we developed a simple template pre-processor¹ which allows sets of entities to be instantiated from generic templates. This has been used to construct all of our data sets, some as large as 140,000 individuals or over 1,000 interconnected locations.

One advantage of this approach, however, is our ability to introduce singular individuals into a simulation with carefully tailored parameters.

Similarly, output from the simulation is also huge, providing a tailorable level of detail on the state of every entity in the simulation over time. The output of a relatively simple simulation, in XML format, can exceed a gigabyte. Analysing such output has proven challenging.

The authors acknowledge that this stands in contrast to the simpler interfaces provided by other simulations. However, we have found in practice that the generality and flexibility of our approach militates against such an approach. We consider the development of better input specification tools to be a research topic in and of itself; a research project is currently underway to develop a output visualisation system that matches the simulation framework in generality and power.

We will now outline recent developments in existing and new modules.

Modules

Improved Dispersion

The dispersion module allows the non-deterministic movement of peeps from cell to cell, according to a probabilistic set of targets. At each tick, each peep within such a cell would be sent to a randomly selected dispersion target.

The dispersion module has been enhanced to unify all existing functionality and introduce new capabilities. These new capabilities are masking, biasing and rate limiting.

Masking allows different dispersion patterns to be applied to peeps based on an individually specified mask. This can be used to have two populations with distinct dispersal patterns moving through the same area and interacting with each other without interfering with each other's movement. For example, consider a football match; each supporter is dispersed from an entry point to the appropriate end of the stadium based on team affiliation, whilst hot dog vendors move through the entire area.

Biasing allows for the relative selection chance of individual dispersion targets to be adjusted. For each dispersion set, each target may be associated with an integer bias value, defaulting to 1. Given a dispersion target T_i from the set of targets T_0, T_1, \dots, T_n , T_i 's chance of selection is given by:

$$P(T_i) = \frac{bias(T_i)}{\sum_{j=0}^n bias(T_j)}$$

¹...which has an unfortunate tendency toward becoming Turing-complete.

Scenario 1, explained later, provides a more comprehensive context for understanding the process of dispersion.

Rate limiting acts to restrict the rates of movement between cells. This is achieved by specifying the period of time between successive dispersions from a given source cell to a specific destination cell. Periods are specified independent of the masked sets and as such are shared amongst them. This ensures that if you impose a limit on the number of peeps allowed to travel between two cells, that limit is enforced irrespective of masking or state. This has the effect of decoupling dispersion rate from tick length.

For example, consider again the football match; rate limiting could be used to model the entrances as choke points that only allow a finite number of people to enter per time period, regardless of their team affiliation.

Motivation Module

The motivation module introduces a mechanism which allows properties of a peep to be affected by the properties of other peeps². The motivation module has three primary functions: to allow peeps to hold *motivators*, to allow these motivators to affect those of other peeps, via *interactions*, and to allow motivators to modify behaviour using *rules*.

This work derives from research conducted into behavioural modelling (Bunder 2010) and is based on an extended and generalised version of the BDI model (Georgeff et al. 1999).

Motivators These are internal state variables which represent things such as belief, attitude, aptitude, aggression, paranoia, hunger, thirst, fear, surprise, efficiency, etc. In other words, each represents a physiological, psychological or sociological parameter. Each motivator has the following elements: *name*, *value*, *resistance* and *influence*. The *name* uniquely identifies a given motivator within the population, for example *charisma* or *panic*. The *value* is the "strength" of the motivator for this peep. It is a real number in the interval $[-1, 1]$. For example, a peep who is completely calm might have a panic motivator value of -1, where as if it was panicking it might have a value of 1. *Resistance*, as the name suggests, is a measure of how much the peep resists change in this motivator. It is a real number in the interval $[0, 1]$ indicating the likelihood that any individual attempt to change a motivator value will fail. *Influence* is the proportion of the population of the cell which the peep can influence in a single time step. This is a real number in $[0, 1]$.

name	Name of the motivator.
value	$[-1, 1]$ value.
resistance	$[0, 1]$ chance to resist changes.
influence	$[0, 1]$ % of local population affected.

Interactions These govern how the motivators of one peep influence the motivators of other peeps in the same cell³. Interactions between peeps are evaluated on a one-on-one basis; thus a peep can successfully influence some fel-

²Environmental effects such as PA announcements can be modelled as static peeps.

³This locality constraint is a fundamental design decision within the simulation framework.

low peeps whilst failing to influence others within the same time step.

Without loss of generality, we will consider the interaction between two peeps. We call the peep who is the source of influence the *skill* while the other is the *rube*.

Each interaction has several elements: *cause*, *effect*, *target* and *strength*. The *cause* is the motivator which induces the interaction. The *effect* is the motivator whose value may be changed by the interaction. The *target*, which has the value “internal” or “external”, indicates whether the peep subject to value modification is the skill (“internal”) or the rube (“external”). Note that this implies that the rube and skill may be the same peep. *Strength* is a multiplicative factor for the overall change in motivator value.

To continue our football example, consider the hot dog salesman (the “skill”) attempting to convince a group of nearby supporters (the “rubes”) of the gastronomic merits of his product; he is attempting to convince them of the belief that what they really want right now is a hot sausage in a bun. The *cause* is “charisma”, the *effect* is “hunger” and the *target* is “external”.

cause	Motivation for interaction.
effect	Motivator being changed.
target	Who will be affected.
strength	Scaling factor.

Rules These are the mechanism by which changes in a peep’s motivators can trigger changes in the peep. Each rule has two elements: a rule set and an action. The rule set is defined in the *rule* attribute; each rule comprises a *motivator name*, a *relation* and a *value*. The possible relations are the common relational operators =, ≠, >, ≥, < and ≤. The value is a real number.

A rule is applied if the value associated with the motivator (m_v), the *relation* operator (\circ) and the rule’s value (r_v) satisfy the expression $m_v \circ r_v$.

For the action to occur, all the rules in the rule set must be satisfied. An action is a set of changes to the rube which may include moving the peep or changing its state. When the rule preconditions are satisfied, the specified action is carried out.

For example, the rule shown in table 1 will cause a peep to change into the “buy a hot dog” state if its “hunger” motivator value is greater than 0.8.

Table 1: Rule Example

Rule	
Motivator	hunger
Relation	≥
Value	0.8
Action	
State	buy a hot dog

In each time step, every peep has a chance to influence some of the other peeps in its cell. For each interaction, the number of peeps to influence is based on the *cause* motivator’s *influence* value and the number of peeps in the cell; the

specific set of *rubes* is randomly chosen. However, if the interaction’s target is “internal”, then the set of rubes consists solely of the skill.

For each selected rube, a $[0, 1]$ random variate is compared to the rube’s *resistance* value for the *effect* motivator; if rejected, no further computation occurs for that rube.

If the rube did not resist, the change in motivator value is calculated from the following equations:

$$e'_r = e_r + \Delta e_r$$

$$\Delta e_r = (e_s - e_r) \times s_i \times c_s \times \frac{\Delta t}{1\text{hour}}$$

Where e'_r is the new value for the rube’s *effect* motivator, e_r is the original value, Δe_r is the change and e_s is the value for the skill. s_i is the strength of the skill’s *interaction* with the rube. c_s is the value of the skill’s *cause* motivator. Δt is the tick length.

Note that this would imply, in our previous example, that the skill would have a high “hunger” motivator value. The vendor will not consume his own stock because he would not have a rule associated with a high hunger value that would permit such an action. If he is capable of this in a different context, the state system would be used to differentiate between these circumstances.

Once all interactions have been computed, rules are evaluated for each peep.

- rules** Rules of the form $m_v \circ r_v$.
- action** Response to rules being satisfied.

A more complex rule set could be used to model this interaction more accurately. In addition to the “hunger” motivator we add the “want hot dog” motivator, being a measure of the desire for a hot dog purchase to occur in order to sate their hunger. The skill will now have a high value for “want hot dog” (he wants them to buy one) and a low value for “hunger” (he knows what’s in them). The corresponding rule shown in table 2 will now induce the purchase of a hot dog provided a peep is both hungry and suitably motivated.

Table 2: Complex Rule Example

Rule	
Motivator	hunger
Relation	≥
Value	0.8
Rule	
Motivator	want hot dog
Relation	≥
Value	0.9
Action	
State	buy a hot dog

On entering the “buy a hot dog” state, we need two things to occur: the hunger and the desire for a hot dog of the peep must diminish and the peep must return to a normal state. This can be accomplished by including a static peep in each cell which is infinitely influential in reducing both hunger

and desire for a hot dog⁴. We also associate a rule with the “buy a hot dog” state which returns the peep to a normal state once their hunger and desire have fallen.

Goal-Based Movement

The goal-based movement module overlays one or more movement graphs over the cells. These graphs associate a number of new properties with each cell. Cells in the graph are one-dimensional paths with physical locations for their end points, a path length and a list of connected paths with their relative locations. Locations along a path are represented as a real number in $[0, 1]$. Note that aside from the proportional location of entities along a path, all values are expressed in quantified, real-world units such as metres and seconds; this is consistent with all appropriate parameters in Simulacron.

Note that we are not attempting, at this time, a full spatial model involving two or more dimensions. We have no reason to believe as yet that the approach we have taken is insufficient for our purposes.

In order to perform goal-based movement, peeps have several additional properties. They must be assigned a speed at which they move along paths. In addition, the module maintains an active path, a relative location within the current cell (provided it is part of the movement graph) and a list of deferred goals.

When a peep is given a goal, either a path to that goal is computed and made active or it is deferred. Goals are deferred when the peep is either already pathing to another goal or the goal is unreachable from their current location. If the peep has no active path, has at least one deferred goal and the first such goal is now reachable, a path to that goal is computed⁵ and is made active.

Scenario 1—Corridor Transit

In order to test and demonstrate the motivation module, a simple proof-of-concept scenario was devised in which we examine the impact of “panic”⁶ on pedestrian traffic flow through a corridor. To this end, we constructed a model whereby peeps enter the corridor at a fixed rate and make their way towards the exit. For consistency, all simulations of the scenario were run with a fixed tick length of 10 seconds for two hours (720 ticks).

Peeps exist in two states: “normal” and “panicked”. In the normal state, peeps in a given cell are subjected to rate limited dispersion biased towards exiting the corridor. In the panicked state, this dispersion pattern is overridden to produce a uniform random choice or direction for each move.

The corridor itself is comprised of a grid of notionally square cells, allowing movement into adjacent neighbours. The corridor entrance and exit are located in the middle and

⁴Similarly, static peeps can be used to gradually increase hunger in the normal state.

⁵This is presently done using the A* algorithm.

⁶Note that we use “panic” in an informal sense. In practice, humans rarely exhibit what we would commonly call panic when subjected to extreme stress. We use it here only because no suitable, concise alternative exists.

at opposing ends of the corridor. Figure 1 shows the layout for the baseline simulation⁷ (which will be described in detail later). The arrows around the peep show movement in calm and panicked states; the relative arrow lengths show relative biases.

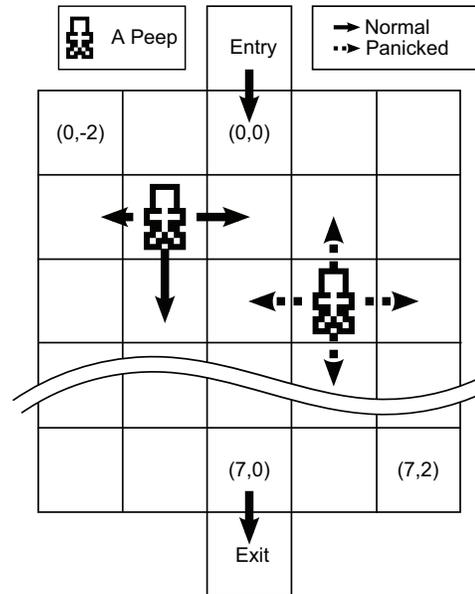


Figure 1: Corridor layout and peep movement

For the first hour of the simulation, there is no panic. At the one hour mark, a single panicking “chicken little” peep is introduced into the corridor at its entrance. The simulation continues for another hour in order to observe what happens. Once this panicking peep induces panic in other peeps, these peeps in turn induce panic in further peeps resulting in propagation of panic.

Each peep was given a single motivator: *panic*. All peeps, with the exception of the isolated “chicken little” peep, begin the simulation in a calm state (*panic* = -1). All peeps are initially in a holding cell which prevents any interaction between peeps. The *panic* motivator has three associated states: *calm*, *panicked* and *flailing*. *Flailing* is merely a stronger, more influential variant of *panicking*. Although the *flailing* and *panicking* states usually have different motivation parameters, they are functionally identical in terms of actual movement behaviour. Unless noted otherwise, “panicking” will be used to refer to peeps in either the *panicking* or *flailing* states.

Once the simulation runs were complete, the output was post-processed to produce graphs of the following derived measures:

Egress Total number of peeps which have left the corridor over time,

⁷Note that the x and y coordinates for the corridor have been transposed in figure 1 for formatting purposes.

Peep states Total number of peeps which are calm, panicked or flailing over time, and

Cell censuses The number of peeps currently in a specific cell over time.

Cell censuses were taken of the cells directly attached to the corridor entrance and exit in order to determine how “crowded” these locations were.

Results

A number of scenarios were simulated, each with slightly different parameters. To wit:

baseline The base simulation from which all others were derived. This used an 8×5 corridor, 5 second period for peep ingress and egress, 10 second period for movement between corridor cells, biases of 5 : 4 : 3 for forward : toward corridor centre : toward corridor edge movement.

The impact of the respective dispersion periods is to permit two peeps to enter the corridor per 10-second tick but only permit one peep to move between any two given cells within the corridor.

The three states, *calm*, *panicked* and *flailing* are associated by rules with the *panic* motivator by the ranges in table 3.

Table 3: Baseline motivator \rightarrow state mapping

State	Range.
<i>calm</i>	$[-1.0, -0.4]$
<i>panicked</i>	$[-0.4, 0.4]$
<i>flailing</i>	$[0.4, 1.0]$

The uniform distributions used for resistance, influence and strength are contained in table 4. The actual values for a given peep are sampled from these distributions. Note that the values used here are selected to produce a panic; they are not derived or modelled from real world behaviour.

Table 4: Baseline interaction distributions

State	Res.	Inf.	Str.
<i>calm</i>	$[0.0, 0.1]$	$[0.0, 0.2]$	$[50, 150]$
<i>panicked</i>	$[0.3, 0.5]$	$[0.2, 0.7]$	$[250, 350]$
<i>flailing</i>	$[0.9, 1.0]$	$[0.9, 1.0]$	$[450, 550]$

diagonal This simulation added diagonal movement between corridor cells.

fastexit This simulation reduced the exit period to 1 second.

fastmove This simulation reduced the corridor movement period to 5 seconds, to match the ingress and egress period.

small This simulation used a smaller 5×3 corridor grid.

weakpanic In this simulation, the interaction strengths were reduced; the new values are contained in table 5.

Table 5: Weak panic interaction distributions

State	Str.
<i>calm</i>	$[25, 50]$
<i>panicked</i>	$[50, 75]$
<i>flailing</i>	$[75, 100]$

superweakpanic This simulation further reduced the interaction strengths, relative to **weakpanic**. The interaction strengths for this run are contained in table 6.

Table 6: Super-weak panic interaction distributions

State	Str.
<i>calm</i>	$[1, 10]$
<i>panicked</i>	$[10, 20]$
<i>flailing</i>	$[20, 30]$

Upon examining the results, most runs followed a similar sequence of events:

1. Peep egress, after a delay of a few ticks, begins and remains at a relatively steady rate.
2. At some point after the introduction of the panicking peep (between 1.1 and 1.3 hours) the egress rate lowers.
3. As the panic spreads, peeps begin to move back through the corridor, eventually causing congestion at the entrance.

The differences between scenarios is in the exact egress rates and time after panic that the rate is affected. Figure 2 shows the egress over time for some of the scenarios. All but one of the runs follow each other very closely until the panic, at which point the egress rates begin to diverge. The exception is the **fastmove** run which has a much higher initial egress rate; this is due to more peeps being able to reach the exit. Table 7 lists the pre- and post-panic egress rates for all scenarios; the values were averaged from five runs each.

Table 7: Pre- and post-panic egress rates in peeps per tick

Run	Pre-Panic μ, σ	Post-Panic μ, σ
baseline	1.143, 0.046	0.609, 0.035
diagonal	1.146, 0.041	0.739, 0.052
fastexit	1.166, 0.048	0.625, 0.049
fastmove	1.671, 0.020	1.150, 0.641
slowexit	0.908, 0.012	0.677, 0.034
small	1.280, 0.027	0.873, 0.285
superweakpanic	1.128, 0.042	1.165, 0.101
weakpanic	1.112, 0.023	0.768, 0.064

Of interest is the behaviour of the **fastmove** and **superweakpanic** runs, where the post-panic egress rate is almost unchanged from the pre-panic rate. They can both be explained the same way: the panic didn’t propagate. In the case of **fastmove**, peeps do not spend enough time in

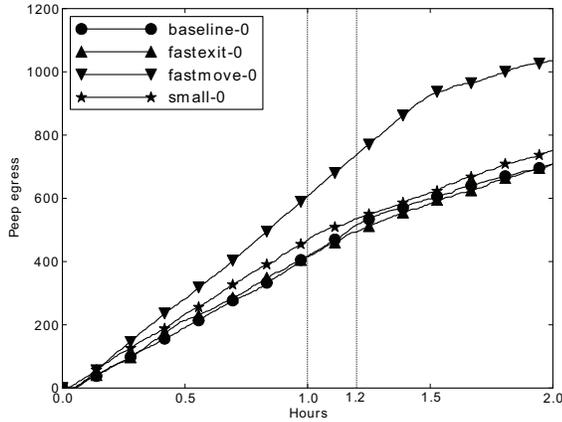


Figure 2: Egress over time

the corridor with already-panicking peeps to become panicked themselves. The reason for **superweakpanic**'s lack of change is more direct: the interactions are so weak that the panicking peep never manages to actually induce a panic in any other peeps.

The spread of the panic also took place largely as expected. Figure 3 shows a graph of the number of peeps in each of the states over time, taken from the **baseline** run (as an exemplar of the general pattern). This particular graph shows three distinct periods in the development of the panic. Prior to about 1.2 hours⁸, although the panicked peep has been injected into the corridor, there is not yet any panic. After 1.2 hours, the number of peeps in the *flailing* state increases steadily as new peeps enter the corridor and become panicked. This curve is as steady as it is due to the number of panicked peeps clustering near the entrance.

Between these two periods, in a narrow band around 1.2 hours, there is a sudden explosion in the number of *flailing* peeps. The reason that there is not slow, gradual growth is that we found the panic *event* to be extremely sensitive. If the interactions were not quite strong enough or peeps did not spend quite enough time in contact with one another, then the panic would not happen at all (this is the case with the **fastmove** and **superweakpanic** runs). If the pre-conditions for a panic *were* met, then the panic would have explosive growth. To capture behaviour more nearly approximating real life, a more complex, multi-parameter model for panic would need to be developed.

The small number of peeps in the *panicked* state likely represents those who have either a high resistance to panic, or who were only minimally influenced. The tapering-off of the *panicked* curve is likely due to all peeps in that state having reached the exit; peeps become effectively “frozen” once leaving the corridor.

⁸Note the zero-suppression of the x-axis. Also note that there clearly cannot be any panic prior to 1 hour as the panicked peep has not yet been introduced.

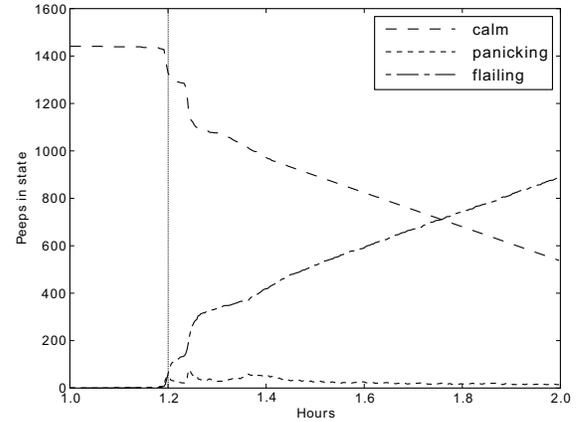


Figure 3: Number of peeps in state over time, **baseline** run

Another movement behaviour which we found interesting was the presence of congestion points. When the panic starts, peeps move about randomly; this eventually causes them to disperse up to the entrance. This in turn causes the population of the cell directly adjacent to the entrance to grow continuously. This agrees with the intuition that if people are attempting to escape and the normal exit is not directly accessible, they may try to get out via the entrance, pushing up against people attempting to enter the corridor.

In addition to the entrance choke-point, there were also two additional congestion points in every simulation: the two corners of the corridor nearest the exit. We believe this is due more to the technique used to model movement than to any property inherent to the scenario. Specifically, peeps which cannot move through the exit due to congestion are likely to eventually move away from it. Unless a peep is panicking, it cannot “backtrack”, thus forcing it out toward the corners. This, combined with the random movement patterns of peeps approaching the exit appears to concentrate the population into these two corners. The dispersion system does not presently track or enforce cell capacities; the ability to enforce cell capacities would strongly mitigate the observed behaviour.

Discussion

We believe that the results of this scenario demonstrate that the new motivation module does work as expected. Although they do not furnish us with any earth-shattering revelations, they do serve as a proof-of-concept for the technique being used.

The scenario does highlight some limitations of using the dispersion mechanism for movement. Chief among these is an inability to model capacity of a location. Logically, what should happen as the corridor fills is that congestion would extend back from the exit toward the entrance, restricting forward movement. This also has implications for behaviour: the inability to move toward their goal might adversely affect a peep's calm, pushing them closer to panic.

This would act cumulatively with the increased time spent in proximity to other panicking peeps.

Additionally, the “move about randomly” behaviour whilst panicking is, while a useful starting point, not terribly sophisticated. Other research within our group suggests that there are a range of behaviours which could arise in these circumstances. Scenario 2 examines these in more detail.

Another potential avenue of improvement would be fine-tuning the movement biases to reflect strategies evident from behavioural observation.

Scenario 2—Transport Hub

We are currently developing a model of the upper level of Central Station in Sydney, Australia to serve as a more complex test bed for these modules⁹.

At the time of writing, this scenario is in its early development stage. The interaction between transport and infection modules has been tested resulting in the successful deployment of 10,000 commuters and the assailant with a subsequent detonation causing 75 deaths and 15 injuries. We anticipate reporting on the full simulation, with the inclusion of the post blast behaviours, in a later presentation.

We are including a brief description of this scenario in this paper to indicate to the reader the size and complexity of the problems we intend to apply the motivation and goal-based movement modules to.

This part of the station has nine entrances and fifteen platforms and serves the intercity rail transport needs of the city. Suburban lines are served from the level below, which is not currently modelled. Approximately one million people move through the station (using both levels) in a day.

There are roughly 150 individual cells in the model, each of which is roughly 10 to 15 meters square, although some (such as platforms) may be larger. The model is run for one and a half hours during the afternoon peak and currently involves the transit of fifteen thousand commuters, each with a distinct entry point and goal platform. This includes the arrival of outbound commuters from a number of entries, each destined for a specific platform and train. It also models the arrivals of commuters on inbound trains and their passage through the station to the exits. The time and numbers simulated correspond closely to the actual traffic levels in the afternoon peak hour. We use real-world model parameters where possible to obtain results which are more readily comparable with observed data. This is in contrast to some other techniques which only use a relatively small, representative population.

An assailant with a backpack IED moves into the station with the intent to kill people awaiting the 4:53 pm train scheduled for platform 9. The assailant has a motivator (“threat”) which influences or motivates others to distance themselves from him.¹⁰

The backpack IED is detonated at 4:51pm wherever the assailant is currently located. Under normal circumstances

this will be somewhere on platform 9. This kills or injures everyone in that cell. At this point, both the terrorist’s level of threat and vitality are seriously impaired.

Subsequent to the detonation a range of behaviours will emerge amongst the other commuters, depending on a complex set of motivators and states. These behaviours range from classical “panic” through attempts to assist to evacuation.

The aim of this simulation is to investigate a number of questions:

- What is the relationship between overt threat signals from the terrorist and the number of casualties subsequent to the blast?
- What are the effects of varying numbers of commuters with specific properties and their strengths? These would include medical training, empathy etc.
- To what extent is behaviour “catching”. This will involve observation of such phenomena as counterflow between those attempting to assist or gain information and those evacuating.

Implementation

The above scenario requires interaction between many of the simulation modules:

- Transport: controls motion of individual peeps from entry to destination.
- Scheduling: controls movement into the transportation network (i.e. the station).
- Dispersion: controls movement out of the transportation network. It also acts to rate-limit movement through choke points (e.g ticket barrier, vending machines).
- Infection: controls the death and injury subsequent to the blast.
- Motivation: controls all behaviours, actions and interactions for peeps other than those already noted.

The most complex of these, for this scenario, is the motivation mechanism which will require significant extension to enable the necessary complexity of interaction and behaviour. This includes a generalisation of the *interaction* mechanism to allow multiple motivators to act in concert to effect change in another. Additionally it requires the evaluation of motivators derived from the peep’s environment. To distinguish these from the motivators already discussed we introduce two new terms: *innate* and *environmental*.

The existing motivators, as discussed above, are now called *innate* motivators to distinguish them from the new environmental motivators.

Environmental motivators are values which are computed from the surrounding environment and used for influencing the peeps. They are intended to be accessed only by a static skill which remains in a single location for the express purpose of influencing all other peeps in that cell. These might include things such as a local lethality index, the number of peeps in a particular state or the maximum value of a particular motivator.

⁹It is interesting to compare this to (Tsai et al. 2011).

¹⁰A feature of eyewitness accounts to the Tavistock bus bomber was that he had a negative influence on people near him prior to detonating the bus (Addley 2011).

In constructing the model, we found the existing, simple interaction mechanism was inadequate. We propose to extend this to allow for arbitrary expressions involving both shill- and rube-derived motivators combined with simple arithmetic operators. We believe this will be sufficiently expressive for our purposes.

This scenario involves eleven motivators of each type; innate and environmental.

The innate motivators are *impatience, training, empathy, sang froid, extroversion, awareness, leadership, threat, threat perception, danger perception and leadership perception*.

The environmental motivators are the number of peeps in each state (ten motivators) and the *proximity to blast* which varies on a per-cell basis.

The behaviours comprise: *normal*, go to platform; *evacuation*, evade the terrorist; *seek information*, move towards the blast site; *catatonia*, remain in place; *hysteria*, move randomly from cell to cell; *assist primary* render medical assistance (requires *training*); *assist secondary*, help render assistance (requires presence of peeps in the ‘assist primary’ state); *normal evacuation*, return to entry point; *evacuation at speed*, move to nearest entry point at a run; and *casualty*, remain in place (injured or dead). This range of behaviours has been observed in post disaster situations such as the London underground bombing of July 2005 (Drury, Cocking, and Reicher 2009).

The choice and structure of the motivators and behaviours is derived from our earlier research (Davies 2010).

Future

In future, as well as further developing the transport scenario, we plan to investigate the application of the motivation system to other, more complex, scenarios.

One such is the investigation of the other end of terrorism: the radicalisation and recruitment of individuals into a terrorist organisation. Here, the subject is being influenced by a charismatic figure who gradually changes their motivators (and consequently, behaviours) over time. Such a model may assist in understanding how the radicalisation process affects individuals, potentially leading to early detection of terrorists within a population.

Another promising potential avenue of research is the use of the motivation mechanism in concert with the infection module to model treatment effects and investigate varying public health policies. This would operate by allowing peeps with appropriate motivators (doctors) to change the infection parameters of other peeps (patients/victims).

The most promising applications of these new mechanisms, then, lie not in their individual capabilities, but rather in their potential, in concert with other mechanisms such as infection or terrorism, to enable the evaluation of policy and strategic decision making.

References

Addley, E. 2011. 7/7 bus bomber jostled passengers with deadly backpack, inquest told. <http://www.guardian.co.uk/uk/2011/jan/12/77-july-7-bomber-inquest>.

Bunder, R. 2010. Making actors “appear” intelligent in microsimulations.

Connor, R. J.; Boer, R.; Prorok, P. C.; and Weed, D. L. 2000. Investigation of design and bias issues in case-control studies of cancer screening using microsimulation. *American Journal of Epidemiology* 151:991–998.

Davies, N. 2010. Study of population behaviours in emergency situations.

Drury, J.; Cocking, C.; and Reicher, S. 2009. Reactions to london bombings. *International Journal of Mass Emergencies and Disasters* 27(1):66–95.

Georgeff, M.; Pell, B.; Pollack, M.; Tambe, M.; and Wooldridge, M. 1999. The belief-desire-intention model of agency. *Lecture Notes in Computer Science* 1555.

Green, A. R.; Piper, I. C.; Keep, D.; and Flaherty, C. J. 2009. Simulations in 3d tactics, interdiction and multiagent modelling. In *Proceedings of 14th Conference on Simulation Technology and Training (SimTecT)*.

Green, A. R.; Piper, I. C.; Keep, D.; Bunder, R.; Davies, N.; and Flaherty, C. 2010a. The application of microsimulation to threat modelling. In *National Security Science and Innovation Conference, 9th Safeguarding Australia*.

Green, A. R.; Zhang, Y.; Piper, I. C.; and Keep, D. 2010b. Application of microsimulation to disease transmission, surveillance and consequence management. In *Workshop on Disease Transmission, Surveillance and Consequence Management held at the University of Melbourne*.

Merz, J. 1991. Microsimulation - a survey of principles, developments and applications. *International Journal of Forecasting* 7(1):77–104.

Orcutt, G. 1957. A new type of socio-economic system. *The Review of Economics and Statistics* 39(2):116–123.

Piper, I. C.; Keep, D.; Green, A. R.; and Zhang, I. 2009. Application of microsimulation to the modelling of terrorist attacks. In *Quantitative Risk Analysis for Security Applications Workshop, IJCAI 09*.

Piper, I. C.; Keep, D.; Green, A. R.; and Zhang, I. 2010. Application of microsimulation to the modelling of epidemics and terrorist attacks. *Informatica* 34:141–150.

Tsai, J.; Fridman, N.; Bowring, E.; Brown, M.; Epstein, B.; Kaminka, G.; Marsella, S.; Ogden, A.; Rika, I.; Sheel, A.; Taylor, M. E.; Wang, X.; Zilka, A.; and Tambe, M. 2011. Escapes - evacuation simulation with children, authorities, parents, emotions and social comparison. *Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* 2(6):457–464.

Weinstein, M. C. 2006. Recent developments in decision—analytic modelling for economic evaluation. *Pharmacoeconomics* 1043–1053.

Zhang, Y.; Green, A. R.; Piper, I. C.; and Keep, D. 2008. Application of microsimulation to epidemic modelling: A case study of the royal naval school greenwich 1920. In *International Symposium on Industrial Safety and Health*.