The application of microsimulation to threat modelling

Tony Green
University of New South Wales

Nicola Davies
University of New South Wales

Chris Flaherty
University of New South Wales

Ian Piper
University of Wollongong, ian@uow.edu.au

Daniel Keep
University of Wollongong, dkeep@uow.edu.au

See next page for additional authors

Publication Details
The application of microsimulation to threat modelling

Abstract
This paper summarises the current research on the use of microsimulation and its use for understanding behaviour in modelling threat in society. The motivation for this project was to develop simulation tools that could be used to model human behaviour and overlay that behaviour with a full spectrum of threats to understand how alternative ways for undertaking prevention, protection and response could be used to reduce the risk from a threat. The basic simulation engine - Simulacron, and its associated modules are briefly described. A number of applications to biological infection, including single vector infectious disease spread, biological weapon placement in a community and multi-vector disease spread are also briefly discussed as examples as to how the basic modelling is developed. While some preliminary work on interdiction of terrorism is discussed, the main thrust of the paper is toward the development of a general model of influence and behaviour that can be utilised with other modules of Simulacron. We believe that this simulation system, even though the tools are still in development, can be applied to a range of terrorism and crime related problems, including acts of terrorism and its indoctrination, the criminal terrorism fusion, criminal gang resource movement, critical infrastructure protection, interdiction and emergency response.

Disciplines
Physical Sciences and Mathematics

Publication Details

Authors
Tony Green, Nicola Davies, Chris Flaherty, Ian Piper, Daniel Keep, and Rachel Bunder

This conference paper is available at Research Online: http://ro.uow.edu.au/infopapers/2517
The Application of Microsimulation to Threat Modelling

*Tony Green*, **Ian Piper**, **Daniel Keep**, **Rachel Bunder**, *Nicola Davies* and Mr. **Flaherty**

*School of Risk and Safety Science
University of New South Wales
a.green@unsw.edu.au

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

21st September 2010

Summary

This paper summarises the current research on the use of microsimulation and its use for understanding behaviour in modelling threat in society. The motivation for this project was to develop simulation tools that could be used to model human behaviour and overlay that behaviour with a full spectrum of threats to understand how alternative ways for undertaking prevention, protection and response could be used to reduce the risk from a threat. The basic simulation engine - *Simulacron*, and its associated modules are briefly described. A number of applications to biological infection, including single vector infectious disease spread, biological weapon placement in a community and multi-vector disease spread are also briefly discussed as examples as to how the basic modelling is developed. While some preliminary work on interdiction of terrorism is discussed, the main thrust of the paper is toward the development of a general model of influence and behaviour that can be utilised with other modules of Simulacron. We believe that this simulation system, even though the tools are still in development, can be applied to a range of terrorism and crime related problems, including acts of terrorism and its indoctrination, the criminal terrorism fusion, criminal gang resource movement, critical infrastructure protection, interdiction and emergency response.

Keywords: Microsimulation, Multi-threat Analysis, Terrorism, Emergency Response

1 Introduction

1.1 Rationale

The motivation of this project is to develop a computer model which allows the modelling of human response to a variety of human threat scenarios. These can range across a spectrum of natural threats such as disease, fire and flood as well as non-natural events including accident and deliberate acts (such as terrorism). The tool is being developed in response to a number of requests from third parties, including government and commercial bodies, to address a wide range of roles, these are:

- **Forensic**: can we gain an understanding of events that have already taken place?
- **Speculative**: can we develop a reasonable "what if" model for potential threats?
- **Decision Support**: can we dynamically model and predict the course of evolving threats?
- **Training**: can we develop realistic models for response personnel training?
- **Evaluative**: can we test the validity of assumptions made by other modelling techniques?
1.2 Microsimulation

Microsimulation is a discrete simulation technique which allows for the modelling of the behaviour of single individuals in a complex system.\textsuperscript{2,3} It was originally devised for financial and economic modelling\textsuperscript{4,5}, but is generally applicable to a wide range of scenarios.

In contrast to traditional statistical modelling, microsimulation involves the modelling of all interactions at the individual level. There is no abstraction. As a result of this, details of behaviour that otherwise would be masked are available for analysis by the researcher.

Each individual in a microsimulation is modelled independently; they are not aggregated in any fashion. This allows behaviours, decisions and properties to be assigned on an individual basis, permitting the introduction of anomalous entities such as disease carriers or terrorists.

1.3 Simulacron

Simulacron is the name of the microsimulation framework we have developed. It abstracts and implements the common functionality needed to build a microsimulation, allowing us to more quickly construct a variety of simulations. Among other things, it handles the simulation's data input, reporting, timing and control. Simulacron was originally designed this way due to the ambiguous nature of our needs at the outset of this project. The design has, however, proven to be a great strength; namely providing us with considerable flexibility.

At the highest level, the most important function of the framework is to provide a mechanism for storing and tracking the state of the simulation. It does this by exposing two main primitives, called ‘cells’ and ‘peeps’, which we define as:

- **Cell**: An abstract location which has no intrinsic properties and whose only purpose is to isolate peeps from one another.
- **Peep**: An abstract individual whose only intrinsic property is its current location (i.e. the cell it is presently contained in).

As already intimated, a forced limitation of the system is that peeps are unable to interact with peeps outside their current cell.

To provide meaning to these two primitives, the framework allows for ‘fields’. Fields are arbitrary objects attached to cells or peeps by name. It also implements a state system allowing fields to have different values depending on the state of the cell or peep to which the field is attached. For example, a peep might have a normal schedule (get up, go to work, come home, sleep) and a different schedule (for instance ‘run around on the spot screaming hysterically’), such as in a panic state.

Another important responsibility for Simulacron is handling the distribution of the simulation across multiple processes using a master/slave architecture, which we prefer to call ‘controller’ and ‘worker’. At present, none of our simulations have been complex enough to require the use of more than a single worker. Regardless, the framework was designed from the outset to be able to easily distribute workloads across multiple workers, taking advantage of the limitation on peeps not being able to interact with peeps in another cell. As well, cells can be arbitrarily divided between workers.

The actual simulation functionality, such as controlling the behaviour of peeps, is implemented entirely through plug-in modules which can be selectively compiled into the final simulation. This allows us to mix-and-match functionality to produce the desired simulation. Rather than require modules to implement their own input routines, the framework handles the loading of the input data sets; these are large XML files detailing every entity in the simulation and their initial parameters. The data set is loaded by the controller and selectively streamed out to the workers based on how the workload is being divided. A reporting mechanism is also provided allowing a user to request that particular reports, at a given frequency
be generated and written to disk. This can be used to either produce a summary of the simulation's activity or a complete transcript of everything that happened during the runtime of the simulation. The reports are written into SQLite databases. This facilitates the generation of graphical and other types of data.

We also plan to introduce a message-passing system in the very near future. This will allow peeps, cells and even external processes to send and receive messages. The initial test case for this functionality will be to support non-localised information sharing such as radio or television broadcasting.

### 1.4 Data Set Template Processor

The input data set for a simulation containing even a trivial number of peeps and cells (on the order of tens of peeps) is so large as to be impractical to write entirely by hand. Although a few programs which generated these data sets from a manageable set of input parameters have been written, we were eventually forced to develop a generalised approach involving data set templates.

By this means, we can specify a generic template for peeps or cells which can then be expanded with the template processor, known as the Data Set Template Processor (DSTP), to produce an arbitrary number of concrete instances. Consider, for example various members of a community, the vast majority of which will be statistically similar, though distinct, within a given amount of variance. In this case, an idealised ‘community member’ may be defined using a template which can subsequently be expanded to populate the rest of the community, whilst still allowing for unique individuals.

### 1.5 Scheduling

By itself, Simulacron imposes no behaviour on peeps, not even movement. Thus, left to their own devices, peeps would simply stay in their initial cell for all time. This is addressed by the scheduling module. Each peep, for instance:

- Has a single active schedule which consists of a sequence of days, each of which is a sequence of schedule events.
- Each event comprises a trigger time (such as 9:00 AM) and an action which can either be an instruction to move to a new cell or to change state.
- The target state may be deterministic or randomly selected from a set.

Implicit in the state change mechanism is the potential for the new state to define a different active schedule. In addition to these cyclic schedules, a peep may also have an ‘event schedule’. Whereas the trigger times for a cyclic schedule are clock times, the trigger times in an event schedule are absolute timestamps; thus, each can only occur once. These can be used to override and cut through cyclic schedules, changing a peep's location or state outside of the normal cycle. For example, it could be used to switch a terrorist from normal day-to-day activities into an active state at a given time and date.

### 2 Application to Infection

The infection module was originally developed to simulate the spread of influenza. Our initial test case was an epidemic which occurred in 1920, at the Royal Naval School in Greenwich, England. This epidemic has been well documented historically, identifying characteristics such as the infection pattern, the social mixing as well as this being a semi-closed community.

The number of peeps and locations was of manageable size for demonstration of the technique. The objective of the study was to demonstrate that the infection module could adequately describe, and match the historical data, recording the actual events involving the spread of the infection throughout the school.

We decided upon a model consisting of the following: susceptible, latent, asymptomatic, symptomatic, heroic and removed. These are illustrated below, which we call the SLASHR ‘states’. Heroic, was a unique
addition which conceptually represents people's tendency to ignore or even conceal sickness. Consider, for example the phrase - it's just a cough, even though this could be a symptom of something far worst. ‘Removed’ can mean the peep has either recovered from the infection and is now immune, or have died.

Figure 1: The SLASHR infection states.

In terms of mechanics, the infection is spread by proximity. A susceptible peep must be in the same cell as an infectious peep. Whether or not the infection is actually passed from an infectious to a susceptible peep is decided on a tick-by-tick basis and is controlled by a random dice roll. This dice roll is affected by the duration of exposure (relative to a normalising base duration) and the individual susceptibility of the uninfected peep.

One problem with using microsimulation as the basis for epidemiological studies is that the literature is geared almost exclusively to traditional statistical methodologies. As a result of this, primary infection parameters are normally unavailable and replaced with derived statistics such as the reproduction rate - ‘R₀’. Sadly, these derived figures provide no direct path to estimation of the actual population parameters of interest to us. In an attempt to remedy this, we have developed an iterative Monte Carlo style technique for this estimation process.

Starting from an initial guess, the program - Refinery, generates a set of data sets sampled from this initial estimation range. It then uses these to iteratively refine the estimation by computing the statistical measures from the output and comparing them to published data. Although convergence is not guaranteed with this process, preliminary results have proved genuinely useful.

By its nature, the output of a single microsimulation run represents one possible outcome from a spectrum of possibilities. As a result, multiple runs are required to fully characterise this spectrum. However, doing this manually is a slow and tedious process. The program - Monotony, was created to automate this process by varying the random seed associated with a fixed data set across multiple runs of the simulation. It also directs the output of each run to a separate file. In the immediate future, it will also perform aggregation and summarisation of results across these multiple runs.

An advantage of microsimulation over traditional simulation techniques is that because every individual is modelled, we can obtain a complete transcript of the progress of the simulation. From this data, we can extract, among other things, a directed graph of infection spread and measure the actual reproductive rate and how it changes during the infection. However, doing this manually is a slow and tedious process (see above). The infection tracer program opens the results database from a simulation run, and computes this directed graph. It can then be output in a number of formats, including the DOT format for input into AT&T's GraphViz package or GEXF which can be processed by Gephi.

Another important mechanism was that of time-dependant isolation. This allowed peeps who were symptomatic to be isolated to a designated location, but only during specific intervals of the day. Consider the case of the Royal Naval School, which was a boarding school. Students showing symptoms were likely to be sent to the school nurse, but this was only during school hours. As well, given the historical period, it was unlikely that any of the teachers would be stalking the dorm rooms at night hunting for potentially sick students. Isolation deals with the heroic state by linearly increasing the peep's chance of detection as they progress from heroic through to fully symptomatic.
It is also possible for cells to become infected, passing the infection on to peeps. This was introduced as a proof-of-concept for modelling infectious agents such as Legionella, which is commonly spread through air conditioning systems; and in this particular case, the building itself becomes sick.

A second case study using an Australian community, based on 6000 houses and 936 workplaces involving 14,000 peeps representing typical demographical distribution matching ABS statistics, uses this capability. However, the initial proof-of-concept occurred before this feature was added. The study involved comparing the detonation of a smallpox bomb, with limited infectivity in four different places within the community. These detonations were random timed to release, from the onset of the simulation. This proof-of-concept study demonstrated that it was possible to get infection spread within the community. One of the surprising results of this study was that infection spread to the community did not occur at all places but only at the more heavily populated locations. As well, a similar number of people were in contact before one became infectious in each of these simulations. Results such as these raise the question as to whether this observation can be linked to a minimum infectious dose that is required to be passed to the individual as observed typically in the literature.

A capability that was added to more accurately model, a more complex infectious agent (and which might answer the above question), was ‘masking’. This allows each peep to have two masks: a group mask, and an infection mask. If enabled, the masking system prevents the infection from spreading between two peeps unless the susceptible peep's group mask and the infectious peep's infection mask overlap. This can be used to model infections that exhibit complex transitions; Schistosomiasis, for example, where the parasite travels from human to faeces, to water-borne miracidia, to snails, to water-borne parasite and back to humans.

Finally, the infection module has a provision for triggering various actions when peeps transition from one infectious state to another. For example, a peep could be placed into a panic state or given a schedule to see a doctor when they become symptomatic.

A third case study involves the SARS outbreak at Amoy Gardens in Hong Kong. This was one of two super-spreaders of infection in Hong Kong, the other being at the Prince of Wales Hospital; suggesting that infection was spread by at least two different vectors. CFD modelling of the flats at Amoy Gardens suggested that aerosol formation from a broken drain and the air currents inside and outside Block E in the complex were the cause of the second vector, although rodents and other possibilities have been mooted (and not ruled out). This problem was chosen for two reasons; to study the effect of forced isolation and to demonstrate a two vector problem could be solved using this technique. A community to represent the group located in Block E with demographic and work characteristics of a typical Hong Kong community was developed for this study. While this latter part of the problem is still being constructed, initial testing shows that:

- The infection spread using WHO accepted time periods for SARS cannot replicate the infection in this building; and,
- Aerosol formation around the rooms in the flat of origin, does have a significant effect on the spread over the entire block community, and hence on assumptions that may be made to control such an outbreak.

### 3 Application to Critical Infrastructure Protection and Terrorism

The Southern Cross Transport hub in Melbourne is being used as an initial test bed for assessing terrorist threat and emergency response in an extended infrastructure that has a regular but variable daily transient population of commuters and tourists. This is an example of a complex space with overlying threats and the research is aimed at developing both normal and emergency response behaviours to different types of terrorist threats.
The development of this has revolved around three modules for Simulacron which are currently in various stages of development and testing:

- The TCP (Terrorist, Civilian, Police) module that characterises security processes;
- The transport module which characterises the connectivity between cells that peeps use for movement; and;
- The AI (Artificial Intelligence) module that provides a sentient response for peeps within the simulation.

3.1 The TCP Module

This module was developed as a proof-of-concept to represent the interaction between insurgents concealed within the general populace and law-enforcement agencies. To this end, the module divides each peep into one of three categories.

- First, there are the **terrorists**; their goal is to remain undetected until they finally attack, killing themselves and all other peeps in the same cell. Aside from their attack time, they have two other parameters: a camouflage factor which represents how adept they are at concealing themselves; and a premature detonation chance which represents their chance to attack upon discovery.
- Second, there are the **police**; and their goal is to locate and stop the terrorists before they can attack. Their only parameter is a perceptiveness factor which represents their ability to "sniff out terrorism." Currently, our model does not include the chance of false positives.
- The final third category is the **citizens**; they have no purpose in the simulation at present, save to keep score.

To date, we have performed a few experiments with this module centred around proof-of-concept^6,7. The module has passed basic sanity checks, namely: increasing police presence reduces detection time; and increasing or decreasing the two factors has the expected effect. While finally, a premature immolation can cause more damage than the planned attack.

3.2 Transport

Although the scheduling module suffices for smaller simulations where it is possible to statically define every peep's movement ahead of time, larger and more complex simulations require a more powerful set of movement primitives. To satisfy this need, we are currently developing a transport module.

The primary mechanism which will be provided by this module is goal-directed movement for peeps. Each peep may indicate a goal cell to the system, which will then move the peep to their goal in a realistic fashion. This involves traversing the interconnected chain of cells between their present location and their desired destination.

Additionally, the system will allow peeps to make use of a combination of various forms of transportation including pedestrians, public and private road vehicles, trains, planes, etc. It will also allow the association of capacity with cells leading to the possibility of congestion and queuing behaviours at chokepoints.

To support this, we are also developing a set of tools which will greatly simplify the specification of the necessary transport networks and associated metadata. This will likely take the form of a compiler which takes vector images specified in the SVG format, extracts the needed information and outputs it as an intermediate format. This intermediate format will then be read in by our template processor and converted into the format understood by Simulacron.

In developing a model of Southern Cross, the proposed test-bed is a simplistic model of a community using a train station. This has been developed for encoding emergency behaviour. The full implementation to the Southern Cross complex is, however, awaiting delivery of the transport module. Figure 2 shows the connectivity of cells in a typical railway station.
Figure 2: Connectivity of cells.

Figure 3 illustrates graphically, some results of simulation for basic population movement from platform to the street through the network represented in Figure 2. The results show normal population movement from the platform to the concourse inside the exit for the station for a community of commuters arriving on two trains fourteen minutes apart during the morning rush. The upper and lower bounds of population have been calculated from two standard deviations away from the mean. The emergency behaviour is being encoded as part of the AI module discussed below.

Figure 3: People moving through three areas for two trains arriving 14 minutes apart.

3.3 AI

In many scenarios sentient peeps pay a crucial role. They are the ones who are making the decisions, performing actions and changing the course of the world; in general, being the mutators of the system. If these sentient peeps do not have the ability to learn and adjust based on their experiences, they are at best equivalent to a robot with a random number generator.

Having peeps that can learn and adapt has many advantages. It gives us the ability to create scenarios that examines such things as how voting intentions change given interaction with other peeps and exposure to media, or human responses to a disaster such as an earthquake or terrorist attack. Situations such as these cannot effectively be modelled without the use of a module that allows for peeps to learn from their surroundings and adapt appropriately.

A review of the literature for emergency evacuation has shown that people react according to influences or cues in their environment as shown in Figure 4. The reaction appears to depend on the degree of perception of imminent danger. If this is perceived as high then a number of different behaviours can be observed which depend on the individual traits of a particular person. This behaviour is different to when the
perception is low, where normal behaviour or seeking information occurs and then only if the strength of this information is strong enough will people evacuate.

Similarly the process of indoctrination, as shown in Figure 5, depends on a charismatic figure or peer pressure that influences a person. Again the susceptibility is dependent on individual traits of the person being influenced.

Figure 5: The process of indoctrination in a community.

The key to making peeps appear intelligent is to allow them to form their own opinions, to be able to share them and eventually to act on them. Take the following example:

‘Alice talks to Bob, telling him about her favourite type of ice cream. Alice's favourite flavours are chocolate and lemon, but Bob is allergic to chocolate.

‘In this case, if Alice is sufficiently persuasive, Bob may listen to Alice and his opinion of lemon ice cream will improve slightly, but his opinion on chocolate ice cream is unlikely to improve due to his medical condition.

‘Carol has a long standing grudge against Alice (which we won't get into here), hears that Alice likes both chocolate and lemon ice cream, so she forms a dislike for these flavours in order to distance herself from Alice.’

The intelligence module adds two new properties to peeps, namely ‘Characteristics’ which hold the mutable properties associated with an individual and may include such things as beliefs and personal attributes such as charisma. While ‘Interactions’ defines how various characteristics interact with each other and how they are passed from one person to the next.
The way in which the Characteristics and Interactions combine is shown in Figure 6. The influencing characteristics are peeps in the environment such as messages or individuals who have persuasive attributes. Group influences such as peer pressure or group think could also be attributed in a similar fashion. The change in behaviour is then predicated on the personal traits such as belief and the resistance to change by a particular characteristic. The higher the resistance the more likely normal behaviour will be observed. Changed actions that result in turn will also have an impact on the influencing environment.

Figure 6: Influence on behaviour.

4 Conclusions and the Future

While the examples shown in this paper are very much a work in progress to develop a suite of simulation tools that are useful for the epidemiological and security communities, we believe that this type of modelling will lead to a basic understanding of population movement and associated behaviours that can be used to test real world problems that are difficult to solve in another way. Multiple threats can in principle be studied in a way that will highlight the effect of human behaviour and the efficacy of government or organisational policy decisions on that behaviour. It is well suited as a technique for forward-backcasting approaches to future prediction of a system through the integration with sensor and surveillance data.

The development of the underlying framework requires demonstration of the integration with other data sources such as simulated hazard data. For example, flood simulations or CFD and finite element simulations of fire and explosion, or chemical dispersion, as well as, including live data such as GIS and surveillance. Additionally some modules are required that better emulate transport, sick buildings and infection spread in people where there are underlying illnesses already existing or where they are made more susceptible to other diseases following infection.

5 Acknowledgements:

This work has been supported by National Science, Security and Technology Unit of Prime Minister and Cabinet, the Australian Research Council through its Linkage program and Arup.
6 References


6) Ian Piper, Daniel Keep Tony Green, Ivy Zhang, Application of Microsimulation to the Modelling of Terrorist Attacks, Quantitative Risk Analysis for Security Application (QRASA), 21st International Joint Conference on Artificial Intelligence (IJCAI), Pasadena, California 11 July, 2009

7) Ian Piper Daniel Keep, Tony Green, Ivy Zhang, Application of Microsimulation to the Modelling of Epidemics and Terrorist Attacks, Informatica, Volume 34, Number 2, 2010, 141-150