Title: Breaking into BIM: Performing Static and Dynamic Security Analysis with the aid of BIM

Authors:

- Stuart Porter, Curtin University, Department of Computing[1]
  Stuart.R.Porter@student.curtin.edu.au
- Terence Tan, Curtin University, Department of Electrical and Computer Engineering[2]
  Terence.Tan@curtin.edu.my
- Dr. Tele Tan, Curtin University, Department of Mechanical Engineering[1]
  T.Tan@curtin.edu.au
- Dr. Geoff West, Curtin University, Department of Spatial Science[1]
  G.West@Curtin.edu.au

Affiliations:

[1]Curtin University, Bentley Campus, *Kent Street, Bentley, Perth, Western Australia 6102*
[2]Curtin University, Sarawak Campus, *CDT 250, 98009 Miri, Sarawak, Malaysia*

Correspondence:

- Contact: Stuart Porter
- E-mail: stuart.r.porter@student.curtin.edu.au
- Mobile: +61438522013
- Postal Address: Stuart Porter, C/O Department of Computing, GPO Box U1987, Perth, Western Australia, 6845

Keywords: *BIM, Security, Simulation, Graph, Multi-Agent, Intelligent Agents, Red Team, Delay*

Highlights:

1. Review the state of BIM and Security Simulation
2. Describe a method for abstracting a graph from a BIM
3. Describe a method of performing security analysis on a graph model
4. Explore use of Multi-Agent Systems for security simulation
5. Demonstrate that Computer Aided Security Simulation provides powerful possibilities for design analysis

Abstract

The design and construction industry is moving towards Building Information Models (BIM) that provide all of the strength of traditional 3D CAD with an added layer of data allowing new and powerful applications. We investigate the concept of using the data within BIM to better explore security design and considerations. We achieve this by first graphing the physical entities of BIM to capture their relational representation as nodes and links. This graph representation will facilitate the use of graph theory or agent-based simulation to assist in the analysis of the static and dynamic behaviour of the environment around the BIM. We describe the implementation of one such graphing method in this paper. We also demonstrate an application of graphing by investigating the use of data within BIM to explore automated infrastructure security design and consideration via red-teaming. The intent is to make security analysis easier and a process that can be carried out during the design phase of a project, even by non-expert users, with useful feedback.

1. Introduction

Physical Security Assessment is the process of examining a facility and establishing the risk of it being penetrated without detection or appropriate response. To achieve this process one traditionally requires a security expert, highly valuable individuals whose knowledge and experience carry a representative cost. This can lead to security considerations becoming almost an afterthought in many cases, implemented as needed with experts often consulted late in a project lifecycle.

Researchers have looked at using computer simulation to assist security practitioners. However, these attempts have often faced problems with the knowledge required by a user to setup and operate the software often impeding the usefulness of the system (Tarr 1992) . In this paper we demonstrate a proof of concept computer aided security simulation tool designed to alleviate these problems by applying known security modelling methods and heuristics to the information contained within a Building Information Model (BIM).

BIM is a 3D modelling paradigm that extends the capabilities of "dumb" modelling applications like traditional 3D Computer Aided Design (CAD) by adding a layer of associated information. By leveraging this information layer it is possible to perform deeper analysis of a facility, such as simulating elements like construction cost and time. In our research we look to provide tools that open up Physical Security as a simulation option.

In this article we will discuss some of the advantages of BIM and existing research on Security Simulation. We will then introduce the process we use to go from BIM to simulation, followed by the static and dynamic simulation applications we have developed to date. Finally we will present the results of our work and discuss where we see it leading in future.

2. Background

In this section we will address, in part, the history of BIM and Infrastructure Security. We will also discuss some of the background of Intelligent Agents, which can be used to further explore the models we create.

2.1 Background – BIM

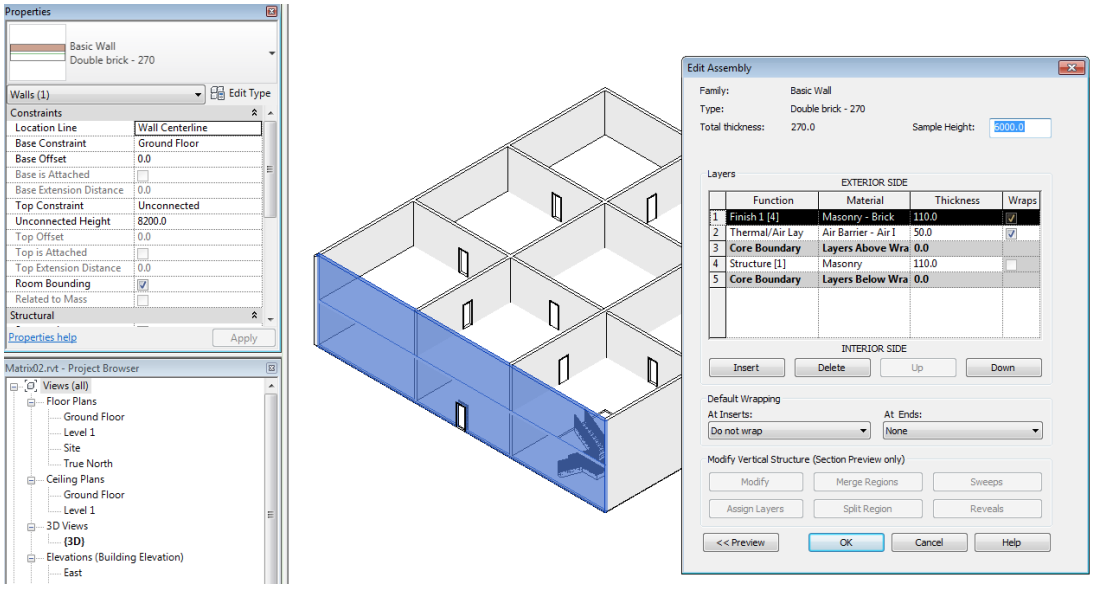

Figure 1: A simple BIM represented in 3D, with the front wall selected

BIM is an evolving standard for the collaboration and design of facilities. We are increasingly seeing its uptake, with many multi-million dollar projects utilising it (Gao & Fischer 2008). In Australia, the federal and state governments are increasingly requiring the use of BIM for public works projects. Many of the organisations using it however are not taking full advantage of its potential.

A common advantage of BIM over traditional CAD is that BIM can perform automatic conflict detection, saving time and money during the design stage. However, Gao and Fischer (2008) found only 14 of the 32 projects they examined took advantage of this feature. Azhar, Hein and Sketo (2007) demonstrated that the returns on investment for BIM from activities such as automatic collision detection are well worth the cost to organisations.

Cheng & Wang (2010) discussed how BIM allows for any changes that need to be made to be implemented more cheaply than in traditional 3D CAD or 2D design methods. One example of this is moving a support beam; In traditional CAD a designer typically needs to move the beam, then any supports, bolts and foundation individually. In BIM these objects can all be grouped, allowing the software to move the associated components with the beam, saving the designer time and effort.

BIM is not without criticism, though this criticism typically lies more with the implementations than the concept. Coates et al. (2010) stated "BIM represents the digital Lego, not the digital clay" referring in part to the inflexibility of current software to accommodate different design methodologies. In their paper, they comment that BIM currently has no real support for free form sketching and other early design techniques. Alternatives are suggested, such as Onuma Planning System, though as BIM continues to mature we will most likely see software vendors attempt to address these issues.

Smith (2009) found a great deal of power in the ability of BIM to help plan and design a facility. They suggested that by better planning how a facility will be used, ensuring loud machinery is away from quiet work areas and so on, the efficiency of a facility can be increased. They go on to state that if the efficiency of a facility can be improved as little as 3.8%, that improvement will pay for the entire facility over its lifetime.

As BIM becomes widely accepted in architectural business, providing tools that make it quick and easy for an architect to receive feedback on their design and bring their attention to areas they may want to adjust will allow for cheap and effective modifications. By building on existing BIM software we intend to make security analysis another tool at the designer's disposal, allowing for easy early consideration and changes to help improve facility integrity. Our tools will not replace security experts but we aim to make some of their knowledge easily accessible, to allow for earlier incorporation of security considerations during the vital design phases when change is easiest to bring about.

2.2 Background – Infrastructure Security

Infrastructure Security is an expansive field in its own right, so here we will look primarily at areas that have influenced our own work. We will first address existing infrastructure security simulation systems. After this, we will provide some background on the security heuristics we have elected to use.

Tarr and Peaty (Tarr 1992; Tarr 1994; Tarr & Peaty 1995) examined the use of computational modelling of security. Their approach examined the use of modelling to assist in the design of prison facilities. To achieve this, a simulation would be setup by modelling all barriers along a given path with relative material strength, which was then analysed to establish if it provided sufficient Delay and Detection.

They found computer-aided simulation to be a beneficial approach, with one of the main limitations being the need for expert users to input data and setup the simulation. A lot of their effort between their original publication and the last was spent on refining the tools to reduce the cost of modelling a facility, but as of last publication it was still a problem. With our system, we have successfully minimised the negative impact of these problems through leveraging BIM.

Others have also used BIM for facility analysis with some also analysing security concerns. In *Automated Assesment of Early Concept Designs* (Eastman 2009), the author describes the work undertaken by their team to create an extension of the Solibri Model Checker. Their extension is reminiscent of an expert system, assessing the BIM design

for conformity to various regulations and providing feedback to the designer, reducing the knowledge burden on designers less experienced in dealing with Courthouse design.

Garcia (2005) described the EASI model, which allows an expert user to calculate the delay along a single path through a facility. Similar to the work by Tarr and Peaty (1995), both the reliance on an expert user and the single path calculation can make this process slow and costly. A point of concern for us is that single path modelling presented in the above systems may lead to the oversight of exploitable attack vectors. The above systems also tend to rely on a user to input delay metrics, increasing their knowledge burden.

To assist in this, we currently use security heuristics based on the work by Alach (2007). Alach used a survey of security experts, academics and students to establish the likely delay matrix for a combination of materials and attacks. While the matrix provides a limited cross-section of all possible materials and attacks, it gives us a good starting basis.

To allow for future adaptability, we have made our system extensible, using extendible Markup Language (XML) formatting, a format that is machine and human readable, for material and attack input so that the system can easily be updated with a users preferred heuristics. The system currently uses a fairly simple name matching system to assign attributes from the XML files to the materials found within the BIM. Once these attributes are assigned, we can then perform a security analysis across the facility.

2.3 Background - Agent based simulation

“The agent can be defined an XML form that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators” (Russell & Norvig, 2003). A multi-agent system is simply an environment that contains more than one agent.

In a multi-agent system, groups of agents interact with each other to achieve some goals. The agent’s individual goal may or may not be aligned with the group’s goal.

The agents have a limited perception and knowledge of the world. Consider that if each agent was to have a complete knowledge of the world, then the behaviour of the system would be akin to a centralized decision maker working through the lower level actuators and sensors (Panait & Luke, 2005).

There are several applications that have shown the benefits of using Multi-Agent Systems (MAS). Industrial applications were developed as early as 1987 in areas including process control, manufacturing and many other areas (Jennings et al. 1995). In defence, MAS has been used in military simulations (McIntosh et al. 2003) (McGrath et al. 2000) and simulating cyberattacks on critical infrastructure (Ganzha & Jain, 2013). This paper introduces the first MAS implementation on BIM for physical security assessment.

3. Method

3.1 BIM to Graph

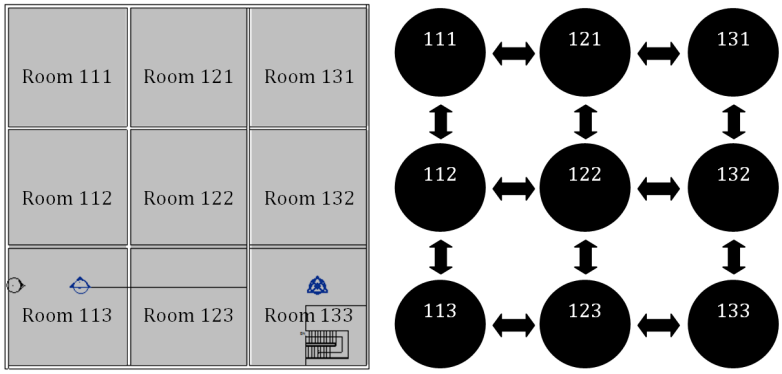

Figure 2: BIM to Graph Transformation

Graph theory is a well studied area in Computer Science and has been used for some time to explore spatial navigation problems. An example of some early work in graph-analysis is presented by Grason (1970), who discusses a system for modelling a building as a graph. As might be expected from early works dealing with computer models, it was limited by the resources available, such as requirements to describe the facility via simple geometry as part of the program.

This is where we and others have found information rich systems such as BIM useful as a basis for graph analysis. Converting a BIM to a graph allows us to take advantage of the knowledge base on graph theory to efficiently explore the problem domain. But to make use of this knowledge base, we first need to extrapolate a graph model from the BIM, a process which we will describe below.

A graph is made up of nodes connected by edges. For security modelling we represent areas or zones, such as rooms, within an environment as the nodes. The edges between the nodes represent the possible paths, with values given to the edges to indicate the cost of that path. Within our modelling, the cost is typically the delay associated with a given path.

In Figure 2 above, we can see a simplified example, with only a single edge between any two nodes. Nodes will frequently have multiple edges connecting them; Doors, walls and windows all provide possible paths between areas. To successfully graph the BIM we must identify all valid paths and generate matching edges, creating a complete graph model of the BIM.

From early exploration of the graph conversion problem, it was decided that generating nodes from raw bounding data from walls would prove too complex and unreliable a process. Working within Revit, Autodesk's BIM authoring solution, we used the Rooms tool to define the areas within a given BIM, taking advantage of the tools intelligence to ensure all areas of interest are labelled. Using the rooms as a basis for our analysis allowed us to search for objects interacting with the bounding box of a given area, reducing our search space and simplifying the problem.

Once the nodes are defined, we then establish the edges between them. As mentioned, potentially any physical link between two areas can provide a path, so we must generate edges corresponding to all walls, doors, windows and even floors and ceilings. In the case of walls, doors and windows, the process is quite repetitive, so we will present here the method for mapping walls, with other paths being quite similar in implementation.

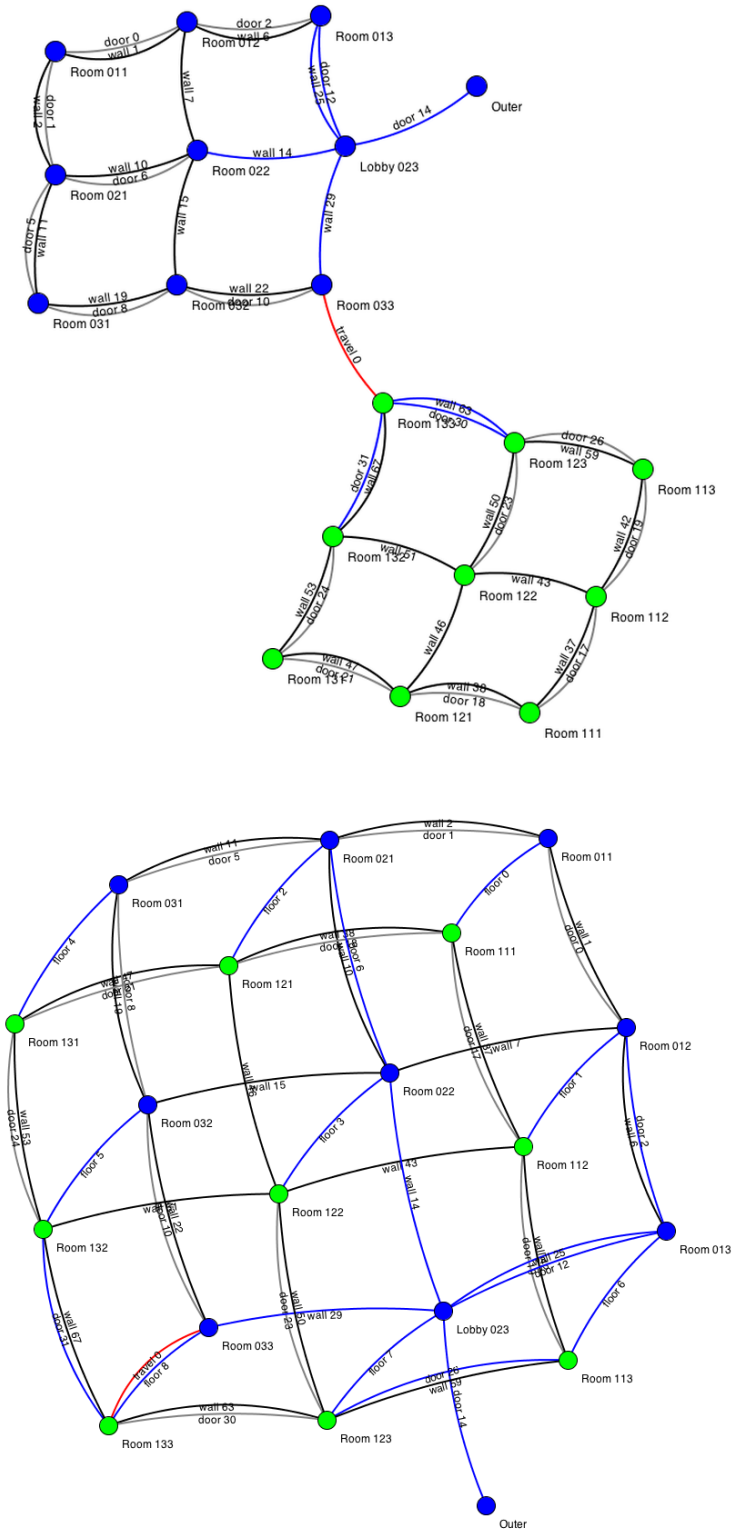

Figure 3: Blue nodes represent ground floor zones, green nodes represent first floor zones. On the top, we have the graph laid out without any vertical links. On the bottom we have added the vertical links.

For each area, we search through all available walls looking for those that intersect the boundary. When such a wall is found, we then search through all other valid areas to test for any that also border the wall and thus are likely to be linked. Due to the inaccuracies of using bounding boxes for this process, we need to check we are not matching across a corner. But if the two zones pass the necessary checks, we create the edge, as can be seen on the top of Figure 3.

In cases where no linked area can be found, we instead create an edge linking to the "outer" zone. These are not shown in Figure 3 to reduce visual complexity. The outer zone is used to represent the world outside the facility and is considered the origin zone for attacks. Once the process is completed for walls, we repeat it for all other horizontal edges, giving us a level-by-level graph representation.

Floors and ceilings are a slightly different process. Because they extend horizontally across multiple zones, we need to use a few different metrics to ensure correct association. Firstly, we go through and establish the boundary heights for each floor of the facility, creating a mapped list of the number of floors and their boundaries. Then for each floor or ceiling, we go through each zone and establish the mid point to test if that falls within the boundaries. If it is within the bounding area, we then check to see if the height of the ceiling or floor would place it adjacent to the zone.

Similar to the process for generating wall edges, we then search through all zones on the level above to identify vertical neighbours. When a neighbour is identified, we create a link between them, as seen on the bottom of Figure 3. For the top-most and bottom-most levels, when no neighbour can be found, we again create a link to the outer zone.

We now have a 3D graph of the facility, taking into account most horizontal and vertical links. The final step is to create the edges representing vertical transport, such as elevators or stairs. As we consider the cost to travel a single floor a subset of the cost to travel a set of floors, we create edges only between neighbouring floors. From a graphing point of view, this greatly reduces the number of edges while still accurately capturing the available paths.

For instance, it is functionally the same in terms of cost when traversing the graph to move from stairwell level one, to stairwell level two, to stairwell level three as it would be to travel from stairwell level one to stairwell level three. By linking level by level we reduce the computational complexity of the graph with no real loss of fidelity.

3.2 Modelling Security – Static Analysis

Security can be considered from the principles of the three Ds; Delay, Detect and Detain. Delay represents the time for an attacker to reach their goal and escape, Detect the probability that they will be discovered along the way and Detain the probability of intercepting them in time. To date, we have been modelling Delay and will discuss below our method for doing so.

We should note that we do not, at present, address the problem of delay due to movement within a room, such as an attacker walking or running. This is due to the complexity of such calculations and unfortunately may not be addressed within the scope of our current research. If time was to allow, we would most likely look at

incorporating a model based upon or similar to the Universal Circulation Network (UCN)(Lee et al. 2010).

UCN attempts to capture more realistic human movement within an environment, rather than simple central line methods where a person is expected to take the most direct route. While for security modelling, a human-like best case navigation would probably give a good indication of Delay effectiveness, it would also be interesting to extend their work. Given time, we could research incorporating attacker familiarity with an area, object clutter and attacker mobility impediment from tool load.

It would also be interesting for larger facilities to incorporate a layered route graph system similar to the work done by Werner et al. (2000). This could allow for fluid assessment of attacker movement within a facility and also attack routes that are applicable to reach an area by car or other modes of transport. For now however we must perform our analysis without this added delay information.

Given the graph generated in Section 3.1, we have a computational searchable representation of a facility. We now assign the edges with appropriate costs, representing the delay involved along a given link. To do so, we consider how an attacker might exploit a given edge by referencing the work by Alach (2007). To make the simulation process as user extensible as possible, the tools and materials are loaded from an XML file.

The XML file contains a break down of all available tools/attack methods. It then lists each material and within that material, a lists of all valid attacks. For each valid attack, an average breach time is recorded in the XML, for use during security simulation.

To perform a security simulation we assign a set of tools to an attacker; such as rock, hammer and explosive. Then, as the graph is generated, we cross reference the assigned tools against the material(s) within a barrier and a link is made between zones using the average breach time(s). In the case of non-material edges, such as stairs, an arbitrary cost value is assigned.

The reason we use an arbitrary value for travel barriers such as stairs is the complexity of properly calculating an actual travel time. The weight of equipment being carried, fitness of an attacker and other factors would affect the actual time. We hope more complex and realistic values can be integrated in the future.

Once all edge costs are assigned, we perform a greedy search. This algorithm traverses the graph, starting from the *outer zone* node with a delay of 0. It finds the cheapest path to each linked node from the *outer zone* and assigns that value to the destination node, as demonstrated in fig 4.

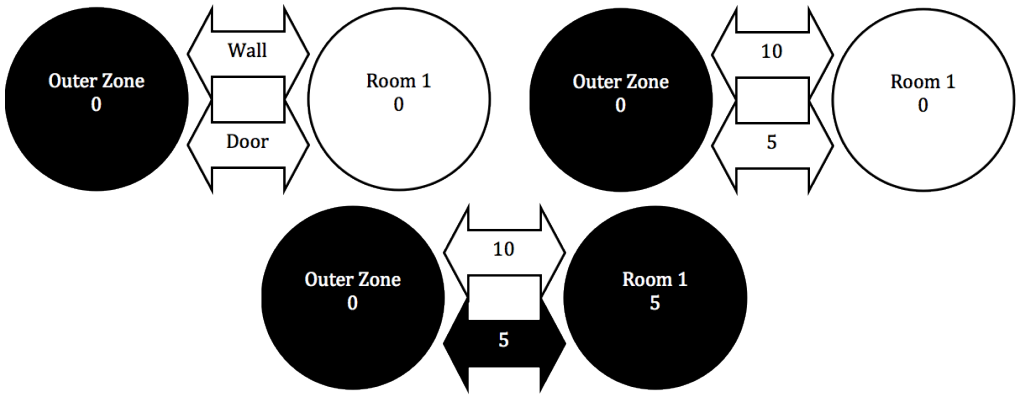

Figure 4: Path analysis example: Select cost for each barrier, Compare costs to establish cheapest path, assign path and value to Room1

The search algorithm then processes each node in sequence, examining its neighbours and adding the lowest link between the neighbours and target node to establish the cheapest path. If this path is cheaper or the node is not yet assigned, it is updated with this value and information on the neighbour responsible. Once it has processed all nodes, the algorithm will repeat the process to allow for changes in cost as nodes are updated. It will continue to iterate over the list of nodes until no further changes occur, indicating the lowest cost to each node has been established.

The above process represents a single simulation run, allowing us to examine the delay to any point in the facility for a given set of tools. To establish common weak points within a facility, the system can perform a comprehensive set of simulations for all combinations of a set of tools. After each individual run, the graph state is saved along with the tools used to reach its stored values. After the final run, a statistical analysis across all states can be conducted, highlighting average delay for each zone and the deviation caused by each tool (see Section 4.1).

3.3 Modelling Security – Dynamic Analysis

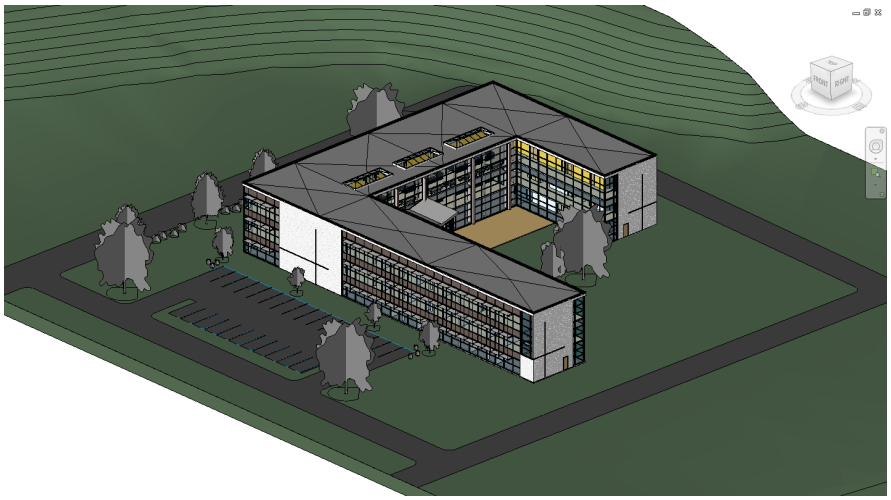

Figure 5: RAC_Advanced_sample_project shown in 3D

The agent-based simulation is developed using the Java Agent Development (JADE) framework for the Multi Agent System and Java Universal Network/Graph (JUNG) framework for the graph representation and visualisation. The simulation uses a hotel model shown in Figure 5.

The agents are divided into two teams: red team and blue team. The red team attacks the facility, while the blue team defends the facility. The red team consists of Red Leader and Red Member, while the blue team consists of Blue Leader and Builder. The toolset is the set of tools available to the agent and are labelled as shown in Table 1. At first, Red Member attacks the facility with a full set of six tools. In each subsequent round, Red Member attacks with one less tool. So Instead of exploring all possible combinations, we simulate a range of attacks: from a fully equipped attacker to a minimally equipped attacker. This reduces the computational complexity but still allow an assessment of toolset's impact on security.

Table 1: Tool set labels

Tool Set	Tools in the set
6	explosive, electric drill, oxy cutting, axe,
5	electric drill, oxy cutting, axe, hammer, rock
4	oxy cutting, axe, hammer, rock
3	axe, hammer, rock
2	hammer, rock
1	rock

At the start, the Red Leader instructs the Red Member to begin the first attack iteration. The instruction contains the attack parameters which are the set of tools, the starting point and the targeted zone, e.g. "Given a hammer and a rock and starting from outsider the building, attack the zone labelled Admin 126". Red Member then computes the shortest path based on Dijkstra's shortest path algorithm (Dijkstra 1959) by comparing the best tool to use and thus the optimum path to take.

Once a room is breached, the Red Member records the time and path taken as well as tools used. An example of this report is shown in Table 2. The Red Member starts again from outside the facility and breaks into the next room and repeats until all rooms are breached.

Next, the Red Member repeats the attack with a different set of tools and submits a final report to the Red Leader. Table 3 shows the time delay for four selected rooms from the 93 rooms available. An operator can conclude that a fully equipped attacker can breach Library 219 in 0.28 minutes whereas a less equipped attacker would take between 4.66 and 22.22 minutes to breach the same room.

Table 2: Detailed log of attack from Outer to Admin 126 using ToolSet 6

Time delay (mins)	Current Zone	Tool used	Material	Barrier breached	Next Zone
0.07	Outer	rock	glass	window 4	Conference 123
0.07	Conference 123	explosive	wood	door 23	Corridor 131
0.07	Corridor 131	explosive	wood	door 27	Admin 126

Once the Red Member submits a final report to the Red Leader, this concludes the attacking round. The report is then submitted to the Blue Leader for further action. This mimics how a red versus blue team scenario would have been conducted.

Now, the Blue team learns from the experience and attempts to harden the facility. By analysing the attack patterns in the report, the Blue Leader instructs the Builder to upgrade a barrier, for example upgrading a door from 'as strong as wood' to 'as strong as brick'. The upgrade strategy for this simulation chooses to upgrade the most frequently used access point first and if it has reached a maximum, it upgrades the next most frequently used point of entry and so on.

Once upgrade is completed, the Builder reports back to the Blue Leader that a barrier has been successfully upgraded. Then, the Blue Leader informs the Red Leader that the facility has been hardened and requests the Red Leader to attack again. This concludes Blue Team's defence iteration. The simulation runs for 100 iterations of attack and defence cycle. The complete simulation takes 20 seconds to execute on a 8Gb 1333MHz DDR3 2.4 GHz Intel Core i7 MacBook Pro.

Table 3: Total delay time for selected rooms and tool set

Tool set	Instruction 315 (mins)	Library 219 (mins)	Lobby 102 (mins)	Lobby 216 (mins)
6	0.60	0.28	0.14	0.21
5	8.52	4.66	3.06	3.13
4	14.44	9.52	6.3	6.37
3	17.78	12.89	9.64	9.71
2	18.30	13.41	9.64	9.71
1	30	22.22	15.42	15.49

Three visualisations views were developed to assist the operator in evaluating the security of the facility. The first view divides the nodes (zones) and edges (barriers) into three equal number groups coloured red (least secure), yellow and green (most secure) as shown in Fig 6. The width of the edge represents the frequency of that path taken. When the simulation runs, the operator can see how each defensive upgrade changes the path taken by the attacker.

The second visualisation as shown in Fig 7 is similar to the first except the threshold is defined by the user instead of the system dividing the graph into three equal parts. This is useful for security audits where there are minimum requirements and where the simulation can immediately highlight zones which pass or fail those requirements. The third view shows the path taken by the attacker from outside the facility to a specified zone as shown in Fig 8. This is useful when a specific zone is of higher interest and the operator wants to view the attack path taken. During the simulation, the attack vector changes in real time to reflect the attacker's responses to the blue team defending actions.

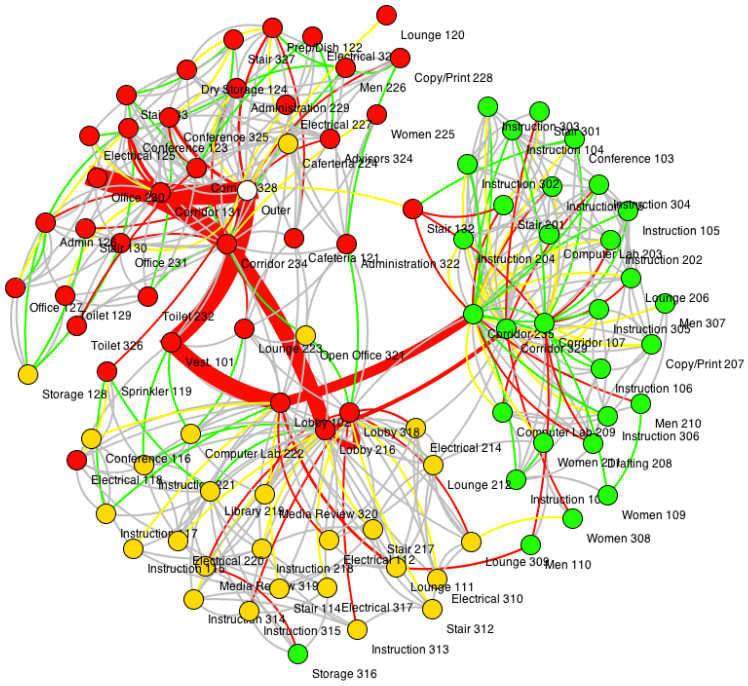

Figure 6: Graph's heat map to show the security levels of the zones

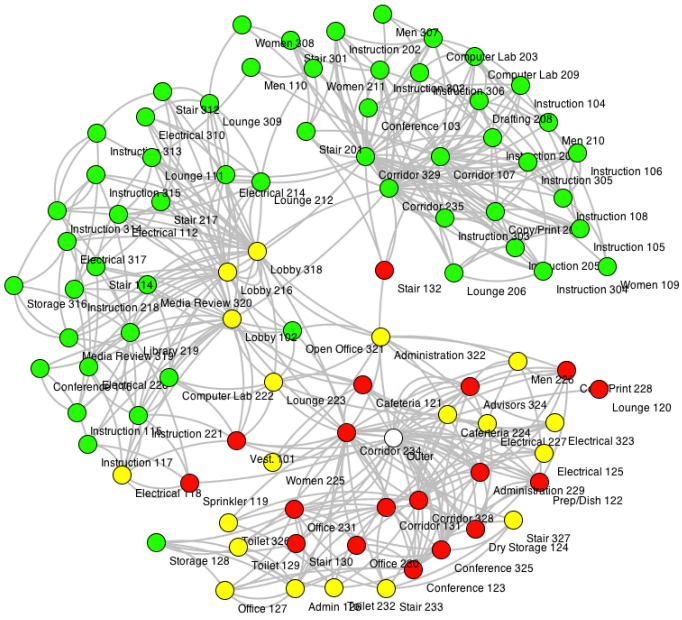

Figure 7: User can define the thresholds of the heat map

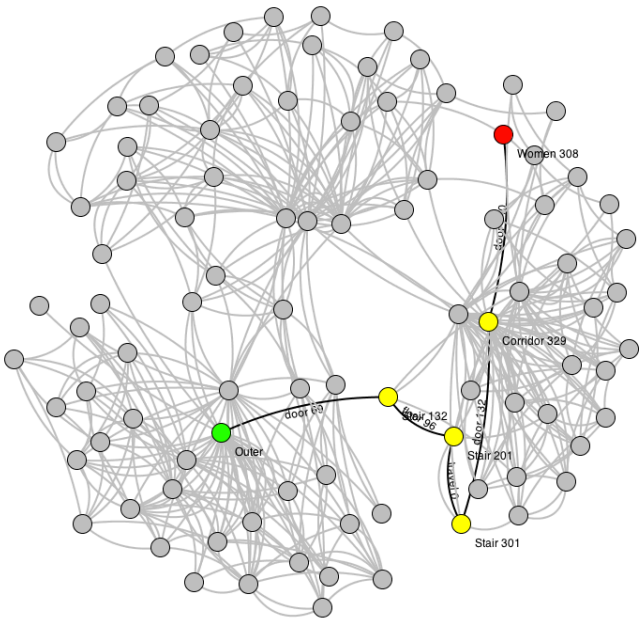

Figure 8: Shows the shortest path from the starting location to the targeted zone

4.1 Results - Static Analysis

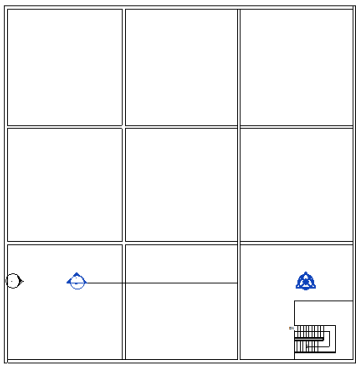

Figure 9: A simple 3x3 matrix BIM of a single floor.

For our development and testing, we created a simple BIM involving two levels of symmetrical rooms. On each floor we created a three by three grid of rooms, as shown in Figure 9. We placed a staircase in the bottom right corner of the grid to link the two floors and used fairly basic materials throughout.

We numbered the rooms based on grid co-ordinates in the form of level, Column and Row. We begin this numbering from floor 0 for the ground floor, with the top left room designated as column 1 and row 1. So in Figure 9, we see the top floor of the simple matrix BIM, where the top left would be denoted as Room 111 and the bottom right would be denoted as 133.

Initial simulations produced the results shown in Figures 10, 11, 12 and 13. The results shown are from a comprehensive analysis of the facility. The different combinations of tools determine the number of simulations. In this case, the tools allow for 63 combinations, giving us a value of 63 for N.

Number of times facility is likely to be breached via this zone

--[Room 011 (131034)] N: 63, Breaches: 32

--[Room 012 (131043)] N: 63, Breaches: 32

--[Room 021 (131037)] N: 63, Breaches: 32

--[Room 022 (131046)] N: 63, Breaches: 0

--[Room 031 (131040)] N: 63, Breaches: 32

--[Room 032 (131066)] N: 63, Breaches: 32

--[Room 013 (13105B)] N: 63, Breaches: 32

--[Lobby 023 (131055)] N: 63, Breaches: 63

Figure 10: First floor breach data for Matrix model

“Number of times facility is likely to be breached via this zone” indicates the number of times that the neighbour used to access the zone is the outer zone. In other words, the number of times it was fastest to access the room directly from outside. Rooms with particularly high breach counts are commonly used for access, indicating some kind of weakness to the attack set used that should be investigated.

As we can see in Figure 10 above, for almost half of the simulations many of the rooms on the ground floor are at risk of being a breach point. These numbers correlate with the runs involving explosives, which have a tendency to heavily weigh the results due to its extremely low time to breach any material. As we would expect, only the lobby which has a door to the outside is used as a breach point in all runs; doors representing a far more exploitable path than a wall.

Against a facility model these results will highlight areas of weakness, allowing a designer to easily examine the common points of attack and improve them, thus improving the overall security of the facility. For instance, in our simple matrix, we can see from the number of breaches that Lobby, bottom centre in Figure 9, is a common entry point. Any improvements to the security of this zone will thus have a trickle on effect to the rest of the facility.

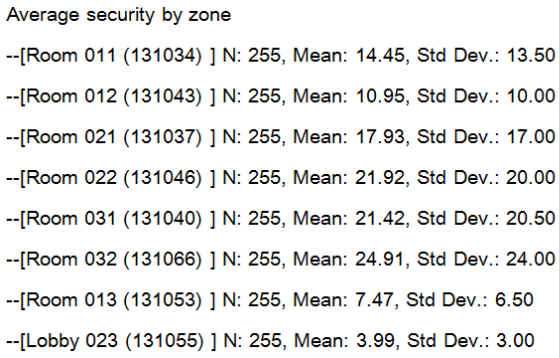

Figure 11: Average delay to each zone and the standard deviation

"Average security by zone" presents the average delay value for each zone across the runs; the time to reach a given zone. A sample output of these values can be seen in Figure 11. As can be seen from the results, the standard deviation is extremely high due to the runs involving explosives. These results are intended to express clearly the likely delay to high value zones, allowing easy evaluation of delay versus response time.

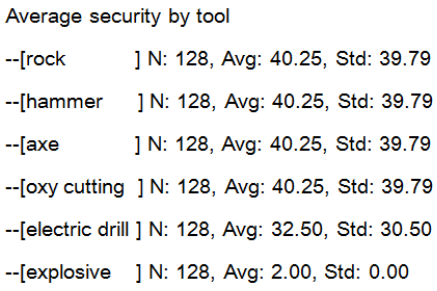

Figure 12: Average delay broken down by tool for the Matrix model

Finally, the system provides the "Average security by tool", breaking down the information for each tool, to allow users to detect attacks they are particularly vulnerable to and attempt to harden a facility against this. The results of this can be seen in Figure 11. We can see clearly from the maximum delay average that Explosives severely limit the security of the given facility, with minimum delay not currently used.

```
Average security by zone
--[Vest 101 (177056)          ] N: 63, Mean: 3.19047619047619, Std Dev.: 2.46149485396573
--[Lobby 102 (177304)         ] N: 63, Mean: 4.85714285714286, Std Dev.: 5.25797094927324
--[Cafeteria 121 (177305)     ] N: 63, Mean: 4.85714285714286, Std Dev.: 5.25797094927324
--[Electrical 118 (177317)    ] N: 63, Mean: 4.85714285714286, Std Dev.: 5.25797094927324
--[Electrical 112 (122324)    ] N: 63, Mean: 6.52380952380952, Std Dev.: 8.18604531312164
--[Lounge 111 (177325)        ] N: 63, Mean: 6.52380952380952, Std Dev.: 8.18604531312164
--[Prep/Dish 122 (177306)     ] N: 63, Mean: 3.19047619047619, Std Dev.: 2.46149485396573
--[Dry Storage 124 (177307)   ] N: 63, Mean: 4.85714285714286, Std Dev.: 5.25797094927324
--[Conference 123 (177309)    ] N: 63, Mean: 3.19047619047619, Std Dev.: 2.46149485396573
--[Electrical 125 (177308)    ] N: 63, Mean: 6.52380952380952, Std Dev.: 8.18604531312164
--[Corridor 131 (177315)      ] N: 63, Mean: 4.85714285714286, Std Dev.: 5.25797094927324
--[Office 127 (177310)        ] N: 63, Mean: 7.55555555555556, Std Dev.: 10.0881476547509
--[Admin 125 (177311)         ] N: 63, Mean: 6.52380952380952, Std Dev.: 8.18604531312164
--[Storage 128 (177312)       ] N: 63, Mean: 9.71428571428571, Std Dev.: 10.5159418985465
--[Toilet 129 [(177313)       ] N: 63, Mean: 6.52380952380952, Std Dev.: 8.18604531312164
--[Stair 130 (177314)         ] N: 63, Mean: 3.19047619047619, Std Dev.: 2.46149485396573
--[Sprinkler 119 (177316)     ] N: 63, Mean: 3.19047619047619, Std Dev.: 2.46149485396573
--[Instruction 117 (177318)   ] N: 63, Mean: 6.52380952380952, Std Dev.: 8.18604531312164
```

Figure 13: Excerpt of average security by zone output for rac_advanced_sample_project

The results shown in Figure 13 are from a run against the "rac_advanced_sample_project" (Figure 5 & Figure 14) that is included as part of the Revit 2012 distribution. To make the project compatible with our toolset some minor changes were necessary, primarily the addition of a few missing zones from hallways to allow proper linking. Placeholders were also added to our material list to allow for matching with materials we did not otherwise have metrics for.

As can be seen from the results, our system is able to deal with more elaborate models as long as some concessions are taken by the designer, such as appropriate naming and rooms throughout. Given these considerations, Cerberus is able to leverage the information within a BIM to provide user readable feedback on the relative security of areas within a facility.

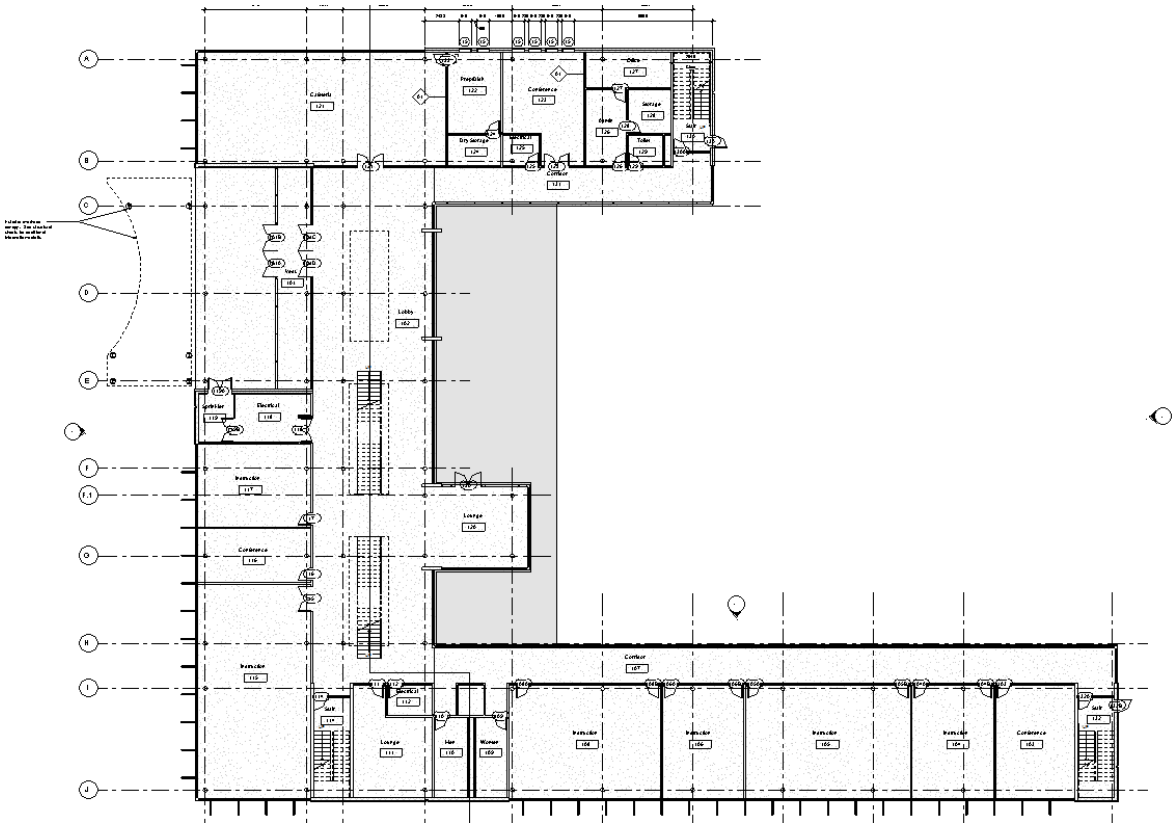

Figure 14: The entry-level floorplan for RAC_Advanced_sample_project

4.2 Results - Agent Based Analysis

Table 4: Number of zones breached within time delay

Delay time (min)	Before	After
t < 5	18	5
5 <= t <10	20	13
10 <= t <15	45	21
15 <= t <20	**9**	**30**
t >= 20	0	**23**

Table 4 shows the number of zones breached in the "rac_advanced_sample_project" model within a certain time. The simulation ends after 100 iterations. In each iteration, Red Team attempts to breach every zone and subsequently Blue Team attempts to harden the facility's defences. The first iteration is labelled as "Before" and the 100th iteration is labelled "After". The number of zones breached in less than 5 minutes has reduced by 13 zones. Before the simulation, nine zones were breached only after 15 minutes. After the simulation has ended, this has increased to 53 zones (shown in bold in Table 4).

At the end of the simulation, the zone with the least delay is Copy/Print 228 (0.64 min) while the zone with highest delay is Women 308 (26.15 min). The average delay for all

zones is 15.12 min. The zone which has a similar delay time to the average is Instruction 117 (15.82 min). Figure 15 shows the gradual increase in the time delay for the zones: Copy/Print 228, Instruction 117, Women 308 and the overall average for all zones.

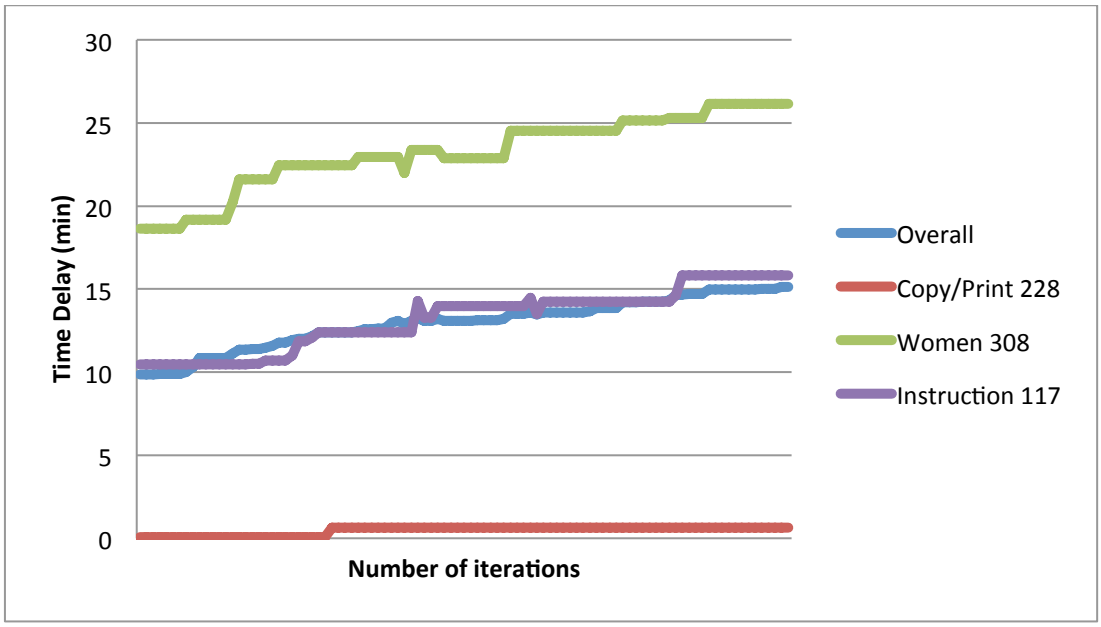

Figure 15: Time delay for selected zones over number of iterations based on the rac_advanced_sample_project model.

The Copy/Print 228 zone has lower security because it is the first entry point from Outer whereas Women 308 zone has higher security because it is on the third floor and separated by infrequently accessed doors.

5. Discussion and Recommendation

We view our work to date as a proof of concept; developing a framework of BIM interrogation and security analysis that can be extended in future work. As our framework matures we anticipate better access to data and more uses for the data we have access to. We will discuss some of the frameworks current shortcomings and future research possibilities below.

An item we would like to see investigated in the future is security-costing estimation. This would allow designers to weigh the cost and benefits of changes they make to security layouts across a facility. Cost estimation is already in use within industry for BIM with regard to materials (Azhar et al. 2007), and appropriate models for estimating security implementation would need to be identified.

There is also potential for using MAS for carbon footprint optimisation. While carbon impact analysis has already been performed on BIM (Stadel et al. 2011) previously, we believe that using MAS to analyse and optimise material selection would be a novel application. Using an expanded framework to export information on wall location and material would make it easy for a MAS to utilise a database and compare for carbon impact of materials as constructed and during lifecycle; similar to the current process for facility hardening.

We are also aware that at present we don't take advantage of all potential information within a BIM with security ramifications, such as walking distance and sightlines. In future we would like to identify and incorporate models for sightlines as well as sound propagation, to enable high fidelity detection modelling. To improve delay modelling, as discussed in section 3.2, we would also like to incorporate a movement modelling system and have already identified some candidates. It will also be important to expand our materials database and begin building a database of objects such as locks and sensors.

The inclusion of sensor detection simulation will also lead to interesting opportunities for the MAS approach. Modelling sensor locations will allow us to perform simulations where attackers must not just reach a destination, but do so undetected, highlighting weak paths. Ideally future iterations would include the ability to have the MAS simulate and recommend best placement for a given set of sensors.

The agent based results show that the BIM can be analysed using a MAS approach. The use of red-blue team is based on simulations in military operations (McIntosh et al. 2003). This approach has not been attempted before using BIMs.

By using a MAS approach, different roles can be developed to provide decision support for the security consultant evaluating a particular facility. A security consultant can map different tasks to different agents to analyse a BIM. For example an agent can monitor and analyse the cost impact of each upgrade in each round.

Security analysis can be further improved by including weights to represent the value of each zone. For example a bank vault would have a higher weight than the men's toilet because it is more important. In future, sensors can be included in the model to simulate detection in addition to the delay aspect.

Other than including additional features in the model, the MAS approach itself can be further improved. One improvement would be to introduce a Blue Member that would either guard a fixed position or patrol selected zones. In future studies, the software can be improved to include multiple Red Members attacking a facility and likewise, multiple Blue Members defending the facility.

6. Conclusion

BIM provides an information rich model, which can be used throughout the lifecycle of a facility, from design through to operation and even decommissioning. Like many things, what you get out of BIM depends on what you put in and organisations that invest the resources in the model will see greater long-term gains. These gains are increasingly driving the design and facility management industries towards BIM, with adoption and innovation within BIM increasing continually.

By building on the basis of BIM, we cannot only value add to those organisations already adopting its use, but we can take advantage of its information layer. This allows us to easily access data necessary to abstract a graph representation of a facility and model its security dynamics. As demonstrated, this allows us to cheaply analyse a facility and provide feedback to the user on the relative security of each zone.

We can then take the generated graph and run agent-based simulation against it. As shown, agent-based analysis can assist in both analysis and design of the facility. This allows for the expedient exploration of a complex matrix of needs and wants, aiding the user in finding their ideal balance.

By providing tools that are easy to use, we intend to facilitate a shift in thinking, allowing security to be considered and incorporated from an early stage, enabling better decisions earlier. We have shown a static and dynamic approach to assess physical security that requires less technical and security expertise than previous work, with minimal cost to implement for an end user. We have established a strong foundation for our future work and demonstrated that these tools can provide powerful analysis to support decisions.

References

Alach, Z. J. (2007). *"Mapping the elements of physical security towards the creation of a holistic physical security model".* Masters thesis, Edith Cowan University, Perth, WA.

Azhar, S. , Hein, M. , and Sketo, B. (2008). "Building information modeling: Benefits, risks and challenges." *Proc., 44th Associated Schools of Construction National Conference* , Auburn, AL.

Cheng, B. and Wang, Y. (2010). "BIM ' s Content And Its Application In Contemporary Architectural Design. In *2010 International Conference on Management and Service Science*, Wuhan, China.

Coates, P, Arayici, Y, Koskela, LJ, Kagioglou, M, Usher, C and O' Reilly, K (2010), The limitations of BIM in the architectural process , in *First International Conference on Sustainable Urbanization*, Hong Kong, China.

Dijkstra, E. W. (1959), "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* In Numerische Mathematik, Vol. 1, No. 1. pp. 269-271

Eastman, C. (2009), "Automated Assessment of Early Concept Designs". Architectural Design, 79: 52–57

Gao, J., and Fischer, M. (2008). "*Framework and Case Studies Comparing Implementations and Impacts of 3D/4D Modeling Across Projects".* Doctoral dissertation, Stanford University, Stanford, California.

Garcia, M. (2005). " Chapter 14 - EASI Computer Model for Analysis ", in Vulnerability assessment of physical protection systems

Grason, J., 1970. "An Approach to Computerized Space Planning Using Graph Theory". In *Proceedings of the 8th Design Automation Workshop,* ACM. New York, NY, pp. 170–179.

Jennings, N., Corera, J. M., and Larsegoiti, I. (1995). "Developing industrial multi-agent system". In *Proceedings of the First International Conference on Multi-Agent Systems* San Francisco, USA, pp. 424-430.

Lee, J.K., Eastman, C., Lee, J., Kannala, M and Jeong, Y.S. (2010). "Computing walking distances within buildings using the universal circulation network". *Environment and Planning. B, Planning and Design*, vol. 37, no. 4, pp.628–645.

Leszczyna, R. (2013). 'Agents in simulation of cyberattacks to evaluate security of critical infrastructures.', in Ganzha, M., and Jain, L. C. (ed.), *Multagent Systems and Applications*, Springer-Verlag, Berlin Heidelberg. pp. 129 - 146.

McGrath, S., Chacòn, D., and Whitebread, K. (2000). "*Intelligent mobile agents in military command and control*" . In *Proceedings of the Workshop on Agents in Industry, Barcelona, Spain*.

McIntosh, G., Galligan, D., Anderson, M. and Lauren, M. (2003): Scythe: Developments in mana agent-based modelling, Naval Postgraduate School, viewed 10 September 2011, <http://harvest.nps.edu/scythe/Issue1/IDFW13-Scythe-Mana.pdf>

Panait, L., and Luke, S. (2005). "Cooperative multi-agent learning: The state of the art". *Autonomous Agents and Multi-Agent Systems*, vol. *11,* no. 3, 387-434.

Russell, S., and Norvig, P. (2003). *Artificial intelligence a modern approach*. (2nd ed.). Prentice Hall. Englewood Cliffs, NJ.

Smith, D. K. (2009). "Ensuring building performance through simulation". *Proceedings of the 2009 Winter Simulation Conference*, Austin, TX.

Stadel, A., Eboli, J., Ryberg, A., Mitchell, J., and Spatari, S. (2011). "Intelligent Sustainable Design: Integration of Carbon Accounting and Building Information Modeling". *Journal of Professional Issues in Engineering Education and Practice*, vol. *137*, no. 2, pp. 51–54.

Tarr, C. (1992). "CLASP: a computerised aid to cost effective perimeter security". *Proceedings 1992 International Carnahan Conference on Security Technology: Crime Countermeasures*. Atlanta, GA.

Tarr, C. (1994). "Cost effective perimeter security". *Proceedings of IEEE International Carnahan Conference on Security Technology CCST-94*. Albuquerque, NM.

Tarr, C. and Peaty, S. (1995). "Using CLASP to assess perimeter security". *Proceedings The Institute of Electrical and Electronics Engineers. 29th Annual 1995 International Carnahan Conference on Security Technology*. Sanderstead, UK.

Werner, S., Krieg-brückner, B. and Herrmann, T. (2000). "Modelling Navigational Knowledge by Route Graphs", in C. Freksa et al., (eds.), *Cognition*, Springer, Berlin Heidelberg, pp.295–316.