School of Electrical Engineering and Computing
Department of Computing

Breaking Into BIM
Physical Security Simulation Utilising Building Information Models

Stuart Ross Porter

This thesis is presented for the Degree of
Master of Philosophy
of
Curtin University

March 2014

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Abstract

Building Information Models (BIMs) are helping to revolutionise the construction and facility management industries. The information layer allows for added intelligence over traditional modelling methods, producing cost savings and value add opportunities which benefit facility shareholders. By exploiting that information layer we intend to create a value add opportunity by making physical security assessment more accessible.

Physical security has traditionally been an expensive process, with many only considering its implications once a facility has reached a near complete state. We propose that a system can be built to allow simulation of physical security from an early design stage, reducing costs and improving implementation. By using BIMs to enable modelling of physical security, we will show the potential for this combination of technologies to help reduce the costs associated with physical security assessment.

To demonstrate this we will devise and demonstrate a system capable of abstracting necessary information from a BIM and formatting it into a edge and node graph model for analysis. Once formatted appropriately, we will be able to exploit knowledge from computer science to analyse the graph and find dynamic properties such as shortest path using Dijkstra's algorithm. The use of these known methods will allow efficient but exhaustive computation supporting automated vulnerability assessment and even virtual red teaming.

This system will draw on knowledge from computer science, simulation, physical security assessment and BIM. By combining these fields we will present a novel solution to the problem by building upon the knowledge from all of them.

The presented solution, while only a proof of concept, will be shown to be functional and capable of performing analysis and assisting a designer in facility testing. Example facilities will be presented beside their graph models to demonstrate the capabilities of the system to model correctly. Security simulations performed on these facilities will then be presented and examined to demonstrate that the system is capable of meaningful output.

This thesis will show that the proposed solution is functional and capable of novel analysis. It will provide details on our methods, reasoning and experimental results. On the strength of these results, we shall also suggest several possible avenues for extension or enrichment of the presented solution.

Acknowledgments

Thank you to Curtin University and the Department of Computing for your support and resources.

Thank you to Tele Tan and Geoff West for supervising me through this process, providing valuable guidance and grounding some of my more mad ambitions.

Thank you to my friends and family for support and the occasional much needed distraction (And a few not so needed).

Thank you to my father for supporting me as always in pursuit of that which I think will make me happy.

Thank you.

Contents

List of Figures

Chapter 1

Introduction

The security sciences deal with attempts to understand vulnerabilities within a design and correct or allow for them. Physical security assessment is undertaken by various means, from purely on-paper analysis requiring inspection, classification and understanding of physical assets and security systems to more active methods such as red teaming, where a group of security practitioners attempt to break into a facility to identify and highlight vulnerabilities. Experts in the field require a specialised set of knowledge to perform their work, which can be referred to as vulnerability assessment, threat assessment, security design or security risk assessment.

The implementation and assessment of physical security is a deep and intricate field of study. While it is grounded primarily in the fields of physical sciences there are also elements of psychology. Proper security analysis requires an assessment of the threats, risks and needs of a facility and its assets. Given these possibilities the consideration of even a simple facility can become a complex task requiring a great deal of time to fully explore.

That time is not cheaply acquired, either. Security experts can command salaries of several hundred dollars an hour, with a small team often being hired for larger jobs where security is paramount. The cost to properly assess a facility can be in the thousands to tens of thousands of dollars before any consideration is given to the cost of recommended changes. These changes may include measures such as personnel training, extra security staff and "hardening" the facility by adding increased physical security if necessary.

Some facilities will self-evidently require this high cost endeavour, such as banks and prisons, to ensure the security of what is contained within them. But with a study of Global Terrorism Database (2012) showing that terrorism rates are on the rise, it can also be prudent to perform vulnerability assessment on facilities where it may seem less intuitive to do so, such as public transport hubs, hospitals and primary resource producers. But doing so using traditional methods can present a cost that companies and governments may find hard to justify.

Computerised security modelling of a facility, however, provides the opportunity to exhaustively model a facility and all possible attack vectors against it quickly and efficiently. Such modelling offers the potential to cheaply explore a facilities vulnerabilities even during early design stages. This approach allows a facility model to be iteratively updated and compared to try and find an optimal design solution.

A primary difference between our work and work by others in the past, such as Tarr and Peaty (1995), Boeing *et al.* (2003), Ustun *et al.* (2006) and Garcia (2001), is the utilisation of Building Information Models (BIMs). BIMs are an emerging architectural technology seeing increased uptake due to a variety of project life cycle benefits. In their paper, Cheng and Wang (2010) discuss some of the advantages as do Azhar *et al.* (2008).

In the past the architectural and building industry were reliant upon 2D drafting using paper, pencil and pens. The industry adopted Computer Aided Design (CAD) as a successor to paper based drafting, allowing for improved sharing and editing. CAD progressed to 3D CAD, allowing a building to be designed in 3D and 2D plans to be produced from that 3D model, improving design speed and accuracy.

BIM was conceived as a substantial step forward from 3D CAD. While it maintains the 3D design element, it uses a database back end to assign information to the objects that are depicted. This means no longer is a pipe simply a collection of lines, but an object which can contain information on its manufacturer, purchase order, PSI limits and anything else that is required.

Amongst its advantages, BIM allows for more expedient editing of models due to intelligent association of objects. It provides better resource estimation, automated clash detection and a reduction in unbudgeted change amongst other financial benefits. The information layer also provides benefits post construction during the facility management phase, where the BIM can be used to store and look up maintenance and part purchase order information in the case of problems. By exploiting this resource we aim to make a security simulation system that is readily useable by industry with a minimum of rework or additional resources.

This research will seek to examine current practices in facility modelling and security simulation. We will then use these practices as a basis to develop a proof of concept system demonstrating the potential for automated vulnerability assessment. We do not expect this system to replace current industry practices but intend it as a supplemental toolset that would potentially allow for cheaper and more robust vulnerability assessment.

1.1 Motivation

Our motivation for undertaking this research is to help improve security within society. We view the cost and knowledge requirements of present day security analysis as impediments to this. So, we seek to use computer science practices to address this through the development of methods that can assist in the process.

In other domains, such as medicine, we have seen expert systems developed that can perform analysis on data to support the decisions of users. If our system can assume a similar role, a expert system for vulnerability assessment, then it could provide support to both expert and non-expert users by reducing domain knowledge requirements and improving analysis. This has the potential to greatly reduce the cost of performing security analysis.

We also know that in most forms of development, the earlier a change is made, the cheaper it is to carry through the process and the better the integration. This is especially true for development of physical media such as buildings, where a change late in the life cycle may well require demolition and construction to modify a facility. If we can develop a toolset which allows for security analysis during the design phase, we can assist in more cheaply and completely integrating security into a facility.

We do not view such a system as a replacement for security experts, but rather as an aide that may be extended to assist security practitioners and lay people alike in quick assessment of a facilities vulnerabilities. Ideally our system could be used by designers to analyse their facility and perform basic hardening from an early design stage. Security practitioners could use the system to help augment their own assessment by providing a exhaustive analysis; Adding confidence to their own results or potentially even highlighting a vulnerability they might have overlooked.

The methods we will use to exploit the information layer within BIM may also prove useful for other forms of building assessment. As such we will look to detail our methods to assist other researchers, as we seek to aid in the development of BIM based value adds.

1.2 Aims

This research aims to provide proof of concept for automated vulnerability assessment. We intend to develop a system that shows the feasibility of performing security simulation on a BIM. Our aims are as follows:

Research and Design
We aim to first identify appropriate toolsets and methods for the development of the system. Whenever possible, we will explore existing related systems and integrate them to develop our own system. This will include work on deciding a appropriate level of abstraction for our security models.

Identify, Extract and Model Data
Secondly, we will aim to identify data available within BIMs that is useful for security modelling. We will explore and develop methods of extracting the necessary data and compiling it in meaningful and accessible formats. We will look to model this data using a graph structure so that we might exploit existing computer science methodologies for exploring the problem space.

Development of Automated Vulnerability Assessment
Thirdly, we aim to explore and develop a system of utilising data extracted from BIM to provide automated vulnerability assessment. We intend for this to demonstrate the viability of incorporating security domain knowledge into a process that can produce useful feedback for non expert users.

Evaluate
The final aim will be to evaluate our methods and provide analysis of their success.

1.3 Problem Description and Scope

Security analysis is a complex task that can help reduce the risk of theft, destruction of property and even terrorism. While it is nearly impossible to eliminate these risks entirely, security analysis can help mitigate the risk and reduce the severity of possible damage. However, as it stands this process is highly knowledge dependant and costly.

To address this, our research will look at developing methods to extract information from

BIMs that can be used for security simulation. This will require identification of appropriate methods for accessing BIMs as well as identifying the information within BIMs that we need to extract. It is also necessary to format the extracted information into a useful form for analysis, such as a graph model.

We will also look at methods for simulating security with computers via computational models. This will require identification of appropriate security models, research into how to implement those models and methods for providing feedback to the user. We will also seek to make our system adaptable so that information can be imported or exported from the system to allow for future tasks such as material additions or virtual red teaming.

The system is intended to provide useful information to users such that security analysis can be performed from even early stages of design. What information is most useful and how we can present it to a user is a difficult problem given our intention that the system should be useable by expert and non expert users, most likely requiring a varied response dependant on user knowledge. Ideally we would resolve this with user consultation and development of effective visual tools for quick information assimilation, but such efforts lie outside the scope of work we're able to address as part of this research and will need to be left for future work.

We envision this system providing benefits in training of security students, supporting security practitioner decisions and informing facility designers.

1.4 Thesis Scope

The scope of our research has been quite broad, as we have endeavoured to encapsulate two disparate areas of knowledge; BIMs and security simulation. Both of these fields cover a large body of work, so within them we have concentrated on the information most relevant to our attempts to integrate the two together. We will present herein the details of our work and the system we designed to demonstrate its potential.

Within the realm of BIM we will examine existing research on the effectiveness of BIM to improve the Architecture, Engineering and Construction industry. We will examine some of the existing ways and means by which researchers have leveraged the information within BIMs to streamline and value add to the design and construction process. Finally, we will present and describe our own methodology for converting a BIM into a usable format for security simulation.

Within the realm of security simulation we will examine previous attempts to automate vulnerability assessment and simulate physical security. While there is a small existing body of work in this realm, it has typically faced difficulties with usability and data input. We hope to overcome some of these difficulties with the assistance of BIM.

We will present the basis for our security simulation and the data that provides our physical material model foundation. We will then discuss our methodology for modelling the elements of security such that they can be used within a computer simulation. Finally, we will present the results of our proof of concept system to demonstrate the potential of this research.

1.5 Thesis Structure

The thesis is structured as follows:

Chapter two is concerned with the context of our research. It will discuss the knowledge and thinking we used as a basis, as well as those which were not informative of our own work but are noteworthy within the relevant fields. We will cover research that relates to, broadly, the categories of BIM and Security Simulation.

Chapter three is concerned with our work with BIMs and extracting data from them. As we will discuss more expansively within the chapter, several toolsets are available for working with BIMs but we selected Autodesk® Revit® for our purposes. We will discuss the criteria for its selection and how we used their API to access the data we needed. We will also discuss the results of our efforts to access the information contained within BIM and organise it into a graph format for simulation purposes.

Chapter four is concerned with our work with security simulation. We will discuss in greater detail the specific security models we used and why. We will also cover in detail how we applied them to the data we retrieved from the BIMs. Finally, we will present the results of this element of our work.

Chapter five is concerned with discussion of the significance of our work to date, our conclusions and possible future work.

Chapter 2

Background and Related Work

In this chapter we shall review research which provides the basis for our own, as well as related works in the area. We have divided the chapter into two primary thematic categories, with some papers which touch on multiple themes necessarily categorised under what we consider their primary theme.

We will look first at Building Information Models (BIMs) and graph theory. BIM is a developing standard within the construction industry and provides a means to convey a deep level of information on a building's design. Graph Theory will be dealt with in relation to buildings and the process to convert a building into a computationally useable representation.

Secondly we will look at Security Research and Simulation. Physical security is a broad and complex field so we will introduce relevant themes and resources to help understand the problem space. We will also examine works within the security field that seek to leverage computational power to assist in security modelling, particularly where it addresses vulnerability assessment.

This chapter will provide an overview of the principles and technologies we will leverage in our work. This should provide for an appreciation of the complexity of the two fields we seek to bring together and set the stage for the later discussions on how we go about achieving our goals. We begin now with a look at Building Information Models and Graph theory.

2.1 Building Information Models and Graph Theory

The concepts of Building Information Models have been around since the 1970s. In one of the earliest papers on the topic, Eastman *et al.* (1974) described a system for using a database to store material information alongside visual geometry. Their system was called Building Description System (BDS) and provides the basis for BIMs as they exist today.

The system allowed for the input of complex graphical elements, a graphical system for editing arrangements of elements and the production of hard copy representations from the computer model. Further, the system was intended to allow management, sorting and manipulation of the database associated with these graphical representations.

The system was developed to address issues raised by traditional methods, typically dependent on drawings. In the traditional system, this dependency leads to problems as a design or building changes over time and the number of drawings that must be catalogued and referenced increases. Traditional paper methods of drawing also require spatial information to be extracted manually from the drawings for any subsequent calculations.

BDS aimed to address these by use of a computer database that contains geometric, spatial and property descriptions. Unlike the traditional system they wished to replace, the database allows for an object to be updated without having to manually update multiple drawings, as all drawings can be generated from the central design. The system was also designed to support component libraries, opening up the opportunity for parts to be described once and used across multiple designs, saving time and effort for designers.

The system was also intentionally general in design. The developers wanted to create a system that would be useful in early design with little regard for imposing a specific design philosophy. They envisioned the system as a general tool that others could extend as necessary.

Since this early work, the BIM platform has developed and matured. Modern BIMs allow for the generation of drawings from models, importing of objects from libraries and data storage as the original BDS system intended. The implementation also commonly includes applications such as automated conflict detection and other advantages over both traditional drawing methods as well as more modern 3D Computer Aided Design (CAD).

Cheng and Wang (2010) discussed in their paper some of the advantages of an organisation using BIMs over more traditional methods such as 3D CAD. As an example they describe the difference between changing a steel beam in a 3D CAD versus performing the same change in a BIM. Within BIM when a user moves a supporting column, the beam will be automatically adjusted to suit because the objects know they are related thanks to the underlying database.

While this might seem trivial, to copy the process within CAD one would need to move the column then delete the extrusion that had represented the beam. One would then need to recreate the line, reintroduce the cross section that represents the shape of the beam and rotate it to match the new orientation. Finally, there is a need to extrude the defined section to recreate the beam.

As it can be seen from the above scenario, BIMs allow for easier adaptation of designs. Not only do they save valuable time in the process, this saved time can be applied to other areas of the design or experimenting with other design ideas. Both of these aspects help to improve the return on investment from BIM design and deployment for the stakeholders.

Other advantages of BIM described include the ability to automate quantity and scheduling updates. Because BIM includes a database layer, various processes that access that layer can be automated. So, in the above scenario with changing a beam, after the beam has been altered the BIM toolset can automatically generate quantity information for ordering purposes accounting for any changes that occurred. An automatic process can also update the building schedule, allowing for timely and up to date information for stakeholders.

While Cheng and Wang (2010) are generally very positive about BIM, even describing it as "the way of the future", they do warn that some care must be taken. They argue that a better definition of BIM is required so that the Architecture, Engineering and Construction (AEC) industry can better understand and mitigate the risks associated with BIM adoption. Cheng and Wang indicate that better definition of attributes such as scope, reference standards, implementation guidelines and business processes amongst others would aid in industry understanding and adoption. For their own efforts, with some dedication and communication they were able to achieve their design goals within BIM and acknowledge its potential going forward.

Smith (2009) stated that if the efficiency of a facility can be improved at least 3.8%, that improvement in efficiency will cover the cost of designing, building and maintaining the entire facility over the course of its lifespan. They argued that these improvements can be achieved by using BIM to simulate the facility. Through this simulation it is possible to reduce worker distractions and optimise facility layout to provide an overall increase in productivity.

An example offered in Smith (2009) is to use the BIM to plan where offices and other facilities will exist within the facility. By examining the location of areas which generate noise, such as plant rooms, the designer can look at how to best arrange the facility to separate high noise areas from areas they would desire to be quiet, such as offices. These improvements increase the efficiency of the people and processes within the building, helping to bring the facility towards optimum productivity.

Smith also discusses the vast costs of facilities; over 40% of all the energy in the United States and 75% of the Earth's raw materials go into producing and maintaining facilities. In this context, facilities are defined as including infrastructure such as roads, dams, bridges and buildings. If restricted to only buildings, we see 40% of raw materials being consumed. When discussing such vast costs, even small improvements in efficiency can have a large potential impact on the cost of a facility.

BIM use within the industry is not yet ubiquitous, but usage is increasing as the advantages are embraced by companies and governments. Gao and Fischer (2008) examined 32 projects which implemented BIM, ranging in scope from several million to several hundred million dollars with both private and public projects examined. They then examined how extensively the projects took advantage of the BIM platform.

To perform their analysis, Gao and Fischer developed a set of "General Beliefs" that they wished to test, concepts considered tacit knowledge to AEC professionals. They then studied the 32 projects, which came from financially and geographically diverse backgrounds and developed a framework for comparison. They then used the framework to help analyse their general beliefs for validity and adjusted those that were found to poorly fit the reality.

An advantage of BIM compared to traditional CAD is automated conflict detection. This allows the software to check objects within the model and highlight instances where two solid objects intersect, a process that must traditionally be carried out either manually at reasonable expense, or corrected for at the time of construction with potentially much greater cost and complexity. They found that in less than half of projects (eg. only 14 of 32 projects) was automated conflict detection used.

This seems to be indicative of BIM usage within industry. Uptake is increasing but usage of advanced features is lagging behind, as Gao and Fischer show, reducing the benefits stakeholders are likely to see from investment in the emerging technology. However, even limited investment in the platform can show dividends, so we expect to see stakeholders continue to expand their usage of available features, improving their return on investment as the platform matures.

In Figure 2.1 we see the analysis of the 32 projects by Gao and Fischer and how they made use of their models. In the graph we can clearly see that models are being used for only some of their potential uses, with many valuable features such as cost estimation used in only a relatively small subset. At the same time, using the model to demonstrate the project to non-professionals was used almost universally.

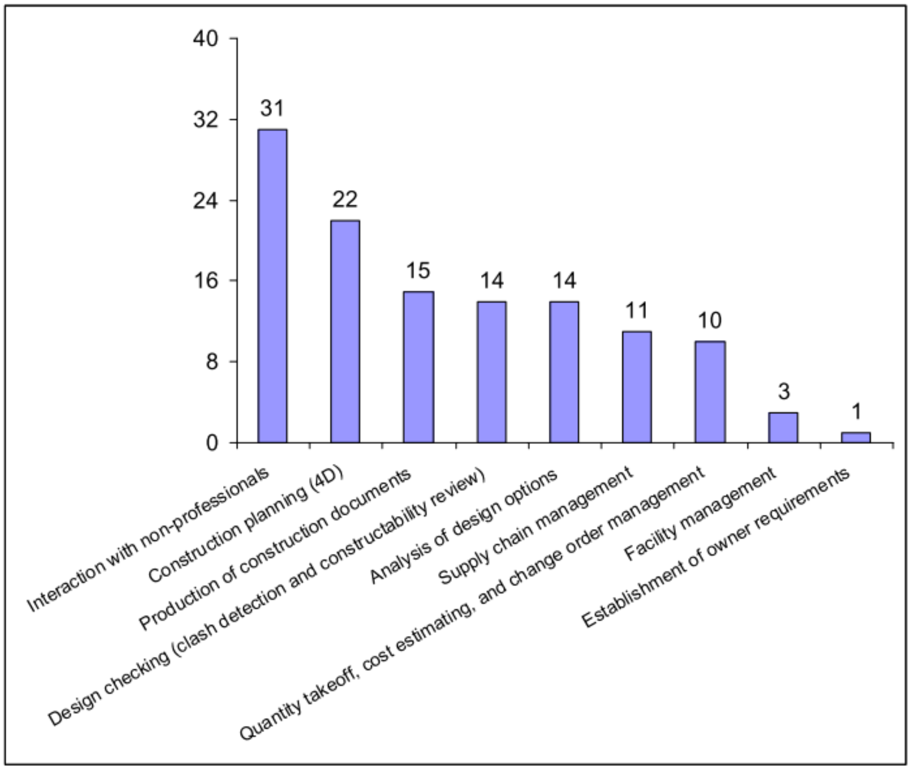


Figure 2.1: A comparison of how models were used within the 32 projects analysed by Gao and Fischer (2008)

Gao and Fischer found that 3D modelling had significant benefits for the design process. It reduced inconsistencies and errors while also reducing the number of design changes required due to a better understanding by the stakeholders. In particular, it seemed beneficial in reducing costly late stage design changes which are often risky due to the number of documents requiring modification and the professionals that must be informed and consulted.

They did not find that 3D modelling necessarily reduced the cost in terms of man hours despite these benefits, but that the effort was reallocated. Time saved on drawing production was instead invested into better quality designs and construction documentation. These changes help to improve client satisfaction, increasing the likelihood of repeat business for the design firm.

In the end, they concluded the following:

- Use of 3D/4D[1] modelling early in the design phase has both immediate and late

---

[1] 4D modelling associates temporal information such as scheduling with the 3D model

term benefits.

- Benefits to the whole team are maximised when all key stakeholders are involved in creating and using the model.
- It was best to create models just-in-time and only to the level of detail necessary for its current use.
- Converting a 3D model to 4D by linking it to the schedule, reviewing the 4D model and updating the 4D model involves technical software issues.
- Reviewing the 4D models and updating them also raised issues with data exchange and organisational alignment.

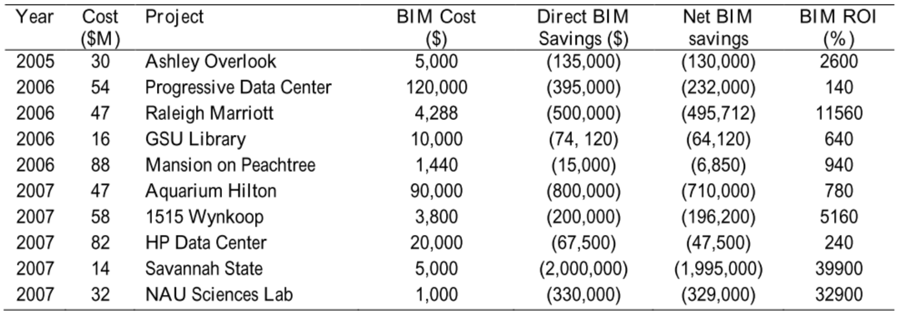
Their paper shows many of the benefits that can be leveraged through the use of BIM and the disappointing utilisation of these benefits in industry at the time of its publication. As utilisation of BIMs within industry increases we would hope to see wider usage of the features it makes possible and this will no doubt be the case. In the meantime their paper shows that BIM is already being applied to a wide range of projects of varying scale to the benefit of the industry.

The work by Azhar *et al.* (2008) further reinforces the benefits of BIM, having examined several other papers on the topic as well as conducting their own research. They present results from other papers which show BIM giving improvements to cost estimation accuracy, reductions in cost from clash detections, reduced project time and up to 40% elimination of unbudgeted changed. They also discussed work that had shown up to 70% of industry respondents were using or considering using BIM, with 75% considering job candidates with BIM skills to have an advantage over those without.

They argue that these figures show that BIM utilisation within the AEC industry is increasing. The papers they reviewed show a large number of benefits on top of those listed above, such as easier sharing of information, better environmental data and improved life cycle data. To help support these qualitative findings, Azhar *et al.* (2008) present the results of a case study they undertook to help quantify the benefits of BIM.

Azhar *et al.* (2008) looked at 10 US based projects ranging from $14 million to $88 million in scope. They found that there was a large spread in reported Return On Investment (ROI), concluding that the variance was most likely due to the varying scope of BIM usage in different projects. While some projects measured BIM savings based on the construction phase, others measured from earlier phases such as the planning or value analysis phase.

While the scope of use may have varied, their data shows that BIM had a positive ROI in all cases. As can be seen in Figure 2.2, the smallest ROI was 140% while the highest was 39,900% with an average ROI of 9486%. While these figures are impressive and certainly suggest a positive return from BIM investment, we should be careful about drawing conclusions from such a small sample size. These values do however only allow for "direct" savings, meaning there is potentially greater life cycle savings on top of these figures, though to what extent is hard to quantify.



Year	Cost ($M)	Project	BIM Cost ($)	Direct BIM Savings ($)	Net BIM savings	BIM ROI (%)
2005	30	Ashley Overlook	5,000	(135,000)	(130,000)	2600
2006	54	Progressive Data Center	120,000	(395,000)	(232,000)	140
2006	47	Raleigh Marriott	4,288	(500,000)	(495,712)	11560
2006	16	GSU Library	10,000	(74, 120)	(64,120)	640
2006	88	Mansion on Peachtree	1,440	(15,000)	(6,850)	940
2007	47	Aquarium Hilton	90,000	(800,000)	(710,000)	780
2007	58	1515 Wynkoop	3,800	(200,000)	(196,200)	5160
2007	82	HP Data Center	20,000	(67,500)	(47,500)	240
2007	14	Savannah State	5,000	(2,000,000)	(1,995,000)	39900
2007	32	NAU Sciences Lab	1,000	(330,000)	(329,000)	32900

Figure 2.2: BIM Economics tables from Azhar et al. *(2008)*

They went on to discuss the risks associated with BIM adoption, highlighting issues relating to legal ownership as well as responsibility for data input. In the first instance, complex issues relating to copyright and licensing can arise and require careful navigation to protect all parties. In the latter, they emphasised the need to negotiate the responsibility associated with ensuring the data in the BIM is input accurately and kept up to date.

Azhar *et al.* (2008) argue that technological and managerial issues are what currently holds BIM adoption back. They highlight a lack of interoperability between software, and even components within BIM, as well as the lack of a clear path for the implementation of BIM, issues which need to be resolved for wider adoption. They concluded that BIM provides lucrative opportunities, but that teams approaching it must beware the pitfalls including the risk that improved cooperation and sharing may eliminate the checks and balances provided by a critical, "adversarial" analysis.

2.1.1 Building to Graph

Some existing work on the conversion of buildings to graphs has been undertaken. Grason (1971) presents an early experimental system for the conversion of a floorplan to a dual linear graph. In this early work they required the use of 11,000 punch cards to describe the program and floorplan to the computer so it might run the conversion, a process that took 23 minutes. One of the floor plans they explored can be seen in Figure 2.3 and describes the ground floor of a building.

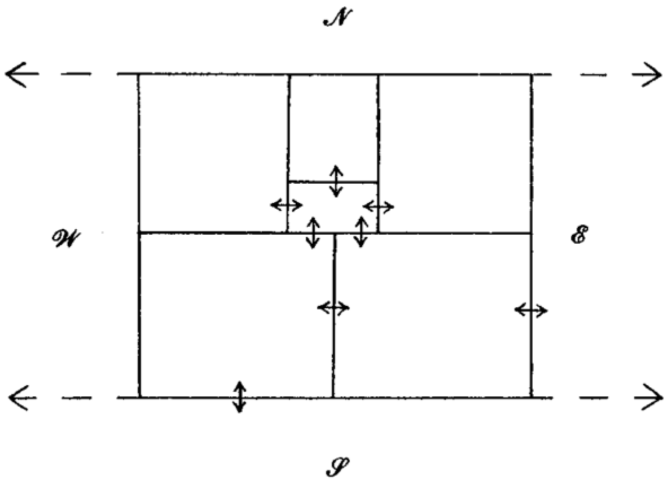


Figure 2.3: A floorplan described by Grason (1971)

The floorplans presented are understandably simple given the processing available at the time. The system also only works with rooms that can be described as rectangles, a limitation most likely attributable to the limited programming capabilities available to them. In the paper they discussed that this limitation can be circumvented to a degree by the use of two or more rooms to create more irregular shapes, while irregular shape buildings can be dealt with by introduction of a "patio“ area to square them off.

The graphing system they developed relies on having four extra rooms numbered sequentially from 01 to 04. These four rooms are defined as N, E, S and W for the cardinal directions of the compass. The four rooms provide a means to reference the external world and better establish how rooms interact.

Grason (1971) described the method by which the program maps each room and identifies the connections between them. In Figure 2.4 we present the completed graph generated by the method. In the figure we can see the four areas designated N through W as the points of the diamond, providing a start point for the analysis.

In Figure 2.4, dotted lines are used to indicate edges that cross north-south or ‘vertical’ walls, while slashed lines indicate edges crossing east-west or ‘horizontal’ walls. The label applied to each edge represents the length of the wall it crosses. While doors are present on the floorplan and restricted to a single door between areas, they do not seem to be

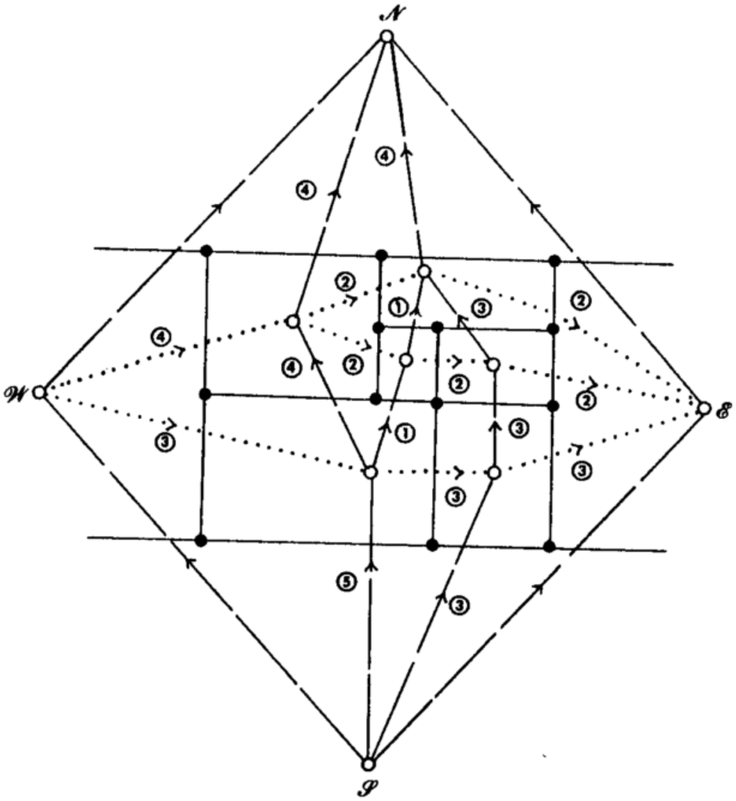


Figure 2.4: The resultant graph from processing the floorplan in Figure 2.3 (Grason, 1971)

represented in the graph by separate edges.

Using the graph and design requirements input at the time of running, the system dubbed GRAMPA (GRAph Manipulating PAckage) (Grason, 1971) was able to perform a series of design activities on the floorplan. GRAMPA was able to test attributes such as well-formedness and provide design solutions that met specified requirements by outputting alternative room arrangements. The system is an early example of using computer modelling and in particular graphing to help establish solutions to physical design problems.

2.2 Security Research and Simulation

In this section we will examine several papers dealing with security research and simulation. We have divided the following works into two rough categories, Security Simulation and Security Metrics.

Security Simulation will deal with works centred around applying security concepts to test a facility design. Security Metrics will deal with works more interested in quantifying rules and data. We will present Security Simulation first below to help illustrate the potential for applied security thinking.

2.2.1 Security Simulation

The work by Eastman (2009) is an interesting case for us to examine as it is one of the few examples combining BIMs and security. While their approach to security analysis is not the same as the one we intend to implement, their thinking none the less provides a certain amount of confidence in our approach. In their work, Eastman and his team looked at ways to utilise the data available within a BIM to perform analysis based on selected standards.

The work undertaken involved the use of Industry Foundation Classes (IFC), an open standard for BIM developed by BuildingSMART[2] to allow greater interoperability and a reduced reliance on proprietary data formats. The models they concerned themselves with are for courthouses within the US; buildings which are required to meet demanding standards and which can be difficult for inexperienced developers to properly assess. They aim to bring this analytical aid to early conceptual models to help maximise project feasibility, stating:

”Further development and detailing later on in the process can refine and elaborate a good early concept, but can only partially ameliorate a bad one.“ (Eastman, 2009, pg 1).

A courthouse in the US is required to meet several standards in order to proceed from concept to construction. These standards range from normal building requirements such as fire regulations to a specific, 400 page set of best practices and requirements laid out in the *US Courts Design Guide*(Vance, 2013). When submitting the design, the architects must also follow the *P-100 Facilities Standards for the Public Buildings Service*(Administration, 2013) design guide, which lists the content required for submission.

Clearly a designer needs to come to grips with a great deal of information to properly design a courthouse and this is a problem Eastman and his team sought to redress. To achieve this end they developed an application suite using Solibri® Model Checker[3] that takes in an IFC file produced by one of the approved software packages. The suite then performs a series of validations on the design, as listed below:

---

[2] http://www.buildingsmart-tech.org/

[3] http://www.solibri.com/products/solibri-model-checker/

- Spatial validation of the layout, comparing target counts and areas of the courthouse project space programme with those of the proposed concept design.
- Circulation analysis of the layout, based on the courthouse-specific criteria of the US Courts Design Guide.
- A preliminary energy assessment, using the Energy-Plus analysis tool.
- A preliminary cost estimate, using the PACES cost-estimating system.

To achieve this design validation via automation the design is required to conform to some additional specifications created by Eastman and his team, such as naming conventions. Before performing the analysis above, a pre-checking review tool first assesses the model for conformity to required standards for various items such as elements, properties and the naming conventions. If the model does not meet the required standards the pre-check tool produces a diagnostic report listing required corrections, thus avoiding unnecessary analysis and helping the process along.

Once the pre-check is passed, the suite begins its spatial analysis. This performs a set of validations created by the team to deal with the level of detail present in an early concept model. The validation produces a report designed to be easily interpreted by an architect, providing information such as efficiency, adequacy, number of spaces and usable area.

Next the suite performs the circulation analysis, which addresses the security assessment aspect. Within a courthouse there are three circulatory areas that must be kept separate; public, restricted and secure. From the US Courts Design Guide the research team identified 216 statements that govern the circulation areas and require validation.

The preliminary nature of a concept design means it is not possible to fully examine the circulatory systems and their interactions as some elements are not fully modelled yet. The suite is still able to perform some analysis, such as validating containment by generating logical sets from the design plans, as seen in Figure 2.5. Adjacent areas with the same security rating are grouped into sets, allowing for quick analysis of interaction between circulatory areas.

Energy analysis is easier to perform on the concept design as it will already include many factors that affect its energy use. These factors include elements such as build materials, skylights, orientation and insulation. Using these factors the energy load required by the design to maintain comfort can be analysed and altered to allow for optimisation.

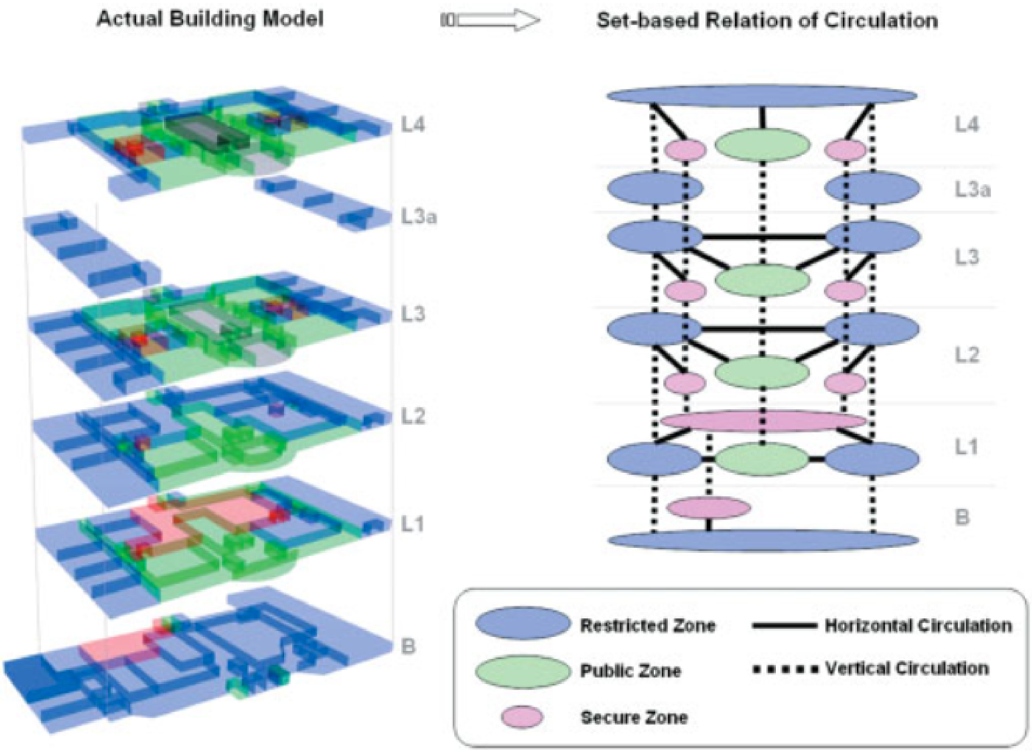


Figure 2.5: An example of a concept design and its circulatory area logical sets from Eastman (2009)

To generate this feedback, certain assumptions are necessarily made by the suite and default values provided to fill in blanks. Thermal load generated by the number of occupants in an area, for instance, are assumed based on the dimensions of the area. Using the provided data and the assumed values, the suite is able to generate a core and perimeter thermal model for user consideration.

Finally, the system performs its cost analysis. This uses a combination of data from the BIM as well as cost-driven data. Cost-driven data includes aspects such as cost-calculation methods, fees, inflation, labour and interest rates.

The conceptual design can then be tracked, using the cost analysis to see how changes to material types and quantities affect the overall cost. This offers the designers the opportunity to better understand the design and, similar to energy analysis, experiment with changes to achieve desired reductions. In combination with other forms of analysis from the suite, when completed, this provides for expedient feedback to the designers enabling better designs and decision making.

While the suite Eastman (2009) has developed is not as security focused as the one we intend to create, it does help demonstrate the increasing usefulness of BIM as a platform. It also shows the potential to exploit the information layer within BIM even at early stages to help inform and shape a design. While their work does not heavily influence our own it is certainly interesting as a demonstration of BIM usage.

Tarr (1992) introduces an early example of computer aided vulnerability assessment. The system described by Tarr was researched on behalf of the Home Office in Britain, a ministerial department responsibility for immigration, security and law and order. The intention was to enable more cost-effective perimeter design for facilities, such as prisons.

In this initial paper, Tarr (1992) described the development of the CLASP system: Computerised Layout Analysis for Secure Perimeters. The system implements a mathematical model for the evaluation of a perimeter cross section, based on the three elements of security, or the three Ds of security. These elements are Delay, Detect and Detain with some practitioners identifying a fourth D; Deter.

- **Delay** deals with barriers that can be presented to slow an escape attempt, such as walls and chain link fences.
- **Detect** deals with alarms, sensors and any guards who might become aware of an escape attempt.
- **Detain** deals with response forces such as guards who may be able to intervene and prevent the escape attempt or recapture the prisoner.
- **Deter** refers to visible security that might discourage an escape attempt at that point within the facility.

Despite acknowledging the importance of deterrence, it is not taken into consideration in their modelling. This is explained by the fact that they do not consider deterrence a pre-requisite for security and, while deterrent elements are in effect in their perimeters, escape attempts still occur. They argue that the effectiveness of a perimeter can therefore be assessed without reference to any potential deterrence, as its other qualities will serve as suitable assessment criteria.

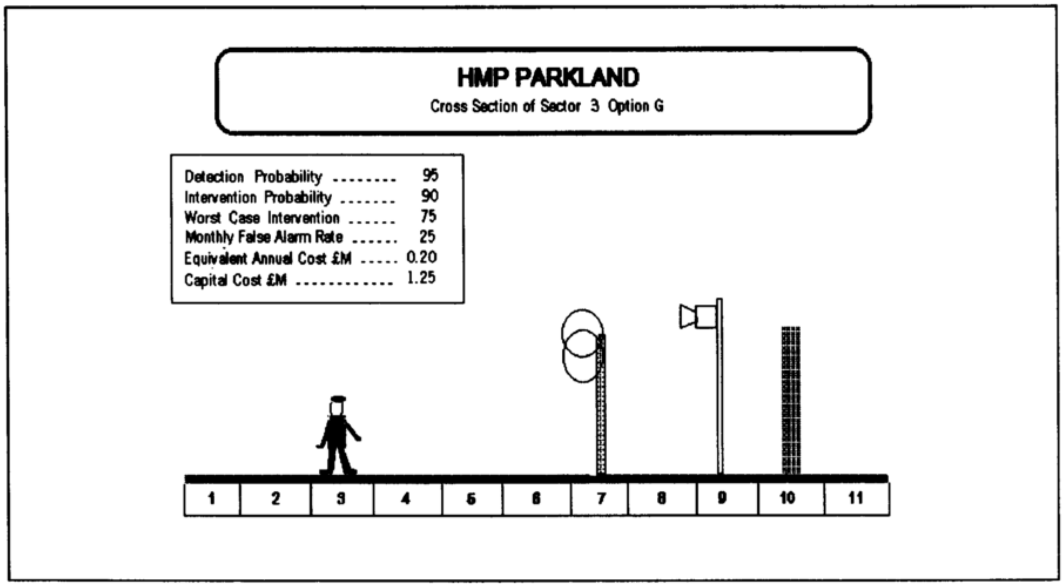
Tarr (1992) describes the evolution of the system, beginning with its initial form, Analysis of Security for Perimeters (ASP). In this initial form, the system required the researchers to implement its modelling functions, meaning it was not widely usable. These early problems lead the researchers to further develop the system to be usable by decision makers.

Their early research identified a number of problems, such as the high volume of data that required input before analysis and the reliance upon an experienced FORTRAN programmer for input and modelling. The system also had little flexibility, limiting its ability to analyse non-uniform cross sections. In response Tarr (1992) went on to create a second iteration called Generalised Analysis of Security for Perimeters (GASP).

GASP improved over its predecessor by adding a menu driven system and an element database; enhancing the ease of information input and reducing the quantity of input required for individual cases. These changes made it possible for non-programmers to use the package and enabled them to add additional items to the element database. GASP also introduced a sector based system, designed to capture variances in the perimeter of a given facility with up to 10 different cross sections programmable.

A cross section was a breakdown of a section of perimeter defence. Within a cross section multiple elements, such as barriers, alarms and responders could be laid out in the layered form they would take in reality. As an example, if you imagine a cross section has ten layers, then the first layer might be a brick wall, the second a chain link fence and the fourth a security guard.

Upon any change to a cross section, the results were automatically recalculated, providing instantaneous feedback to the user. An example cross section can be seen in Figure 2.6. With these improved capabilities over its predecessor, GASP was used to analyse the existing perimeter policy for prison facilities.



HMP PARKLAND

Cross Section of Sector 3 Option G

```
Detection Probability ........      95
Intervention Probability ......     90
Worst Case Intervention ......      75
Monthly False Alarm Rate ......     25
Equivalent Annual Cost £M .....   0.20
Capital Cost £M ...........       1.25
```

Figure 2.6: An example of the cross section screen from GASP (Tarr, 1992)

The research showed that while the existing security policy could be improved upon, it could only be done at considerable cost. It was also found during their analysis of existing facilities that in spite of implementing the same perimeter policies, some facilities were considerably more secure than others due to peculiarities in the perimeter layout or local weather. GASP was therefore still useful to help analyse scenarios where the standard perimeter policy could not be implemented due to local conditions.

The research team identified a few ways in which GASP was still lacking, such as problems with usability and a lack of flexibility. Some options were difficult to change once set, or applied across all options within a sector in spite of changes to a given cross section. These problems promoted the development of the third and final iteration, CLASP.

CLASP's most important change was the move to a layout based design, allowing a user to work with elements across sectors to ensure compatibility. Feedback was improved to be more immediate, similar to ASP, with any changes to a perimeter ranked and displayed in a list, allowing fast and easy comparison to its alternatives. The number of sectors, perimeter shape and 'effective' perimeter were all also made adjustable without recourse to tedious repetition or re-entry.

In the end, Tarr (1992) concluded that analytical models such as CLASP were better suited to operational use than simulation modelling. They did stress that with any model it is important for a user to understand the basis for the calculations as well as be aware of any simplification used within the model. In their next paper, Tarr (1994) go on to describe the system after it has been in use for approximately two years and the findings in that period.

They found that, despite its promise, CLASP was not being used by security practitioners on the scale they expected. Various possible sources for this disconnect between expectation and use are explored in the paper, such as issues with the user interface and practitioner resistance to change. However, they found that these reasons did not seem to account for the shortfall in use.

They surmised that there were two main factors in the limited success of the software. The first factor they considered relevant was the time investment required by the practitioners to see useful returns from the software. The second factor they believed to be relevant was the lack of practitioner confidence in the model.

The model, being simplified, cannot represent all the factors that might affect a system in reality. This is compounded by the practitioners being unfamiliar with the modelling process, effectively turning the CLASP system into a black box with little ability for the practitioners to understand the underlying process and thus the result. Tarr (1994) chose to look at ways to better integrate CLASP into the decision making process by identifying opportunities to better apply the model within the decision process.

Tarr (1994) go on to discuss various theories and models of decision making presented in the literature so that they might examine how the process works and postulate how CLASP might be better integrated. They discuss how analytical models are derived from objective reasoning, as only objective reasoning can be specified, but that this does not mean subjective reasoning is not important. In Figure 2.7 they show the shifting burden between objective and subjective reasoning as tasks become more complex and argue that to improve acceptance of the CLASP system the two sides must be treated as complementary, and not competitive.

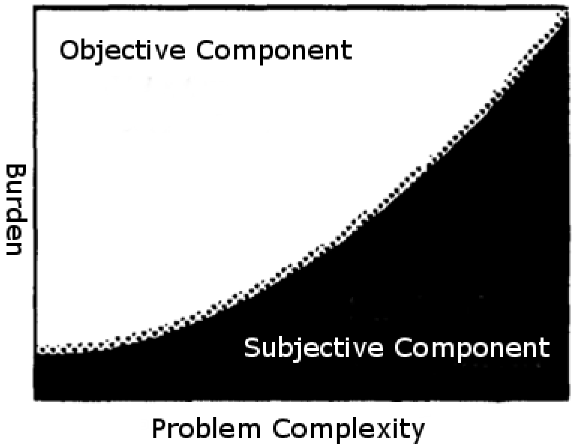


Figure 2.7: Components of Decision Making: Objective Component vs Subjective Component from Tarr (1994). New labels added for clarity

They discuss some of the weaknesses of subjective reasoning, such as bias weighting solutions towards recent problems. They discuss how the objective analysis of the CLASP system can support this. In this way the two systems can serve as compliments, objective reasoning supporting the weaknesses in subjective to ensure better overall solutions if practitioners can be encouraged in its use.

To improve practitioner acceptance of the system, the researchers devised a graded system. The intent behind this system was to make CLASP more granular, adding levels of complexity only when a practitioner was comfortable with the current level. By this method they intended to improve practitioner familiarity with the underlying mechanisms within CLASP so that they could more comfortably rely on its output.

To achieve this, ten levels of practitioner grades were devised. The researchers emphasise in their paper that progression should not be forced, that instead practitioners should only progress when they feel comfortable. The grades were designed to ensure each level provided useful analytical support for a practitioner so that CLASP's benefits might be maximised, even if it's not being fully utilised.

In their final paper on the system, Tarr and Peaty (1995) present a use case for the CLASP system. They used CLASP to examine the perimeter of a low security prison, with CLASP being utilised to compare the current arrangements to possible alternatives. They state that for a standard use case such as this, it is unlikely that a drastically different perimeter will be produced but the process will allow for greater confidence in the resultant perimeter.

They proceed to walk through the steps of performing the analysis. The first step is to model the existing facility, a representation of which can be seen in Figure 2.8. The facility must be broken down into sectors and modelled as accurately as possible, with sectors determined by commonality of perimeter conditions and security qualities such as response time.

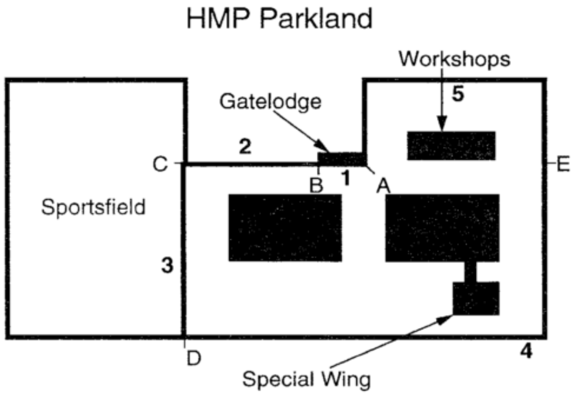


Figure 2.8: An example facility used in Tarr and Peaty (1995)

Once broken down into sectors, the elements within the sector must be input. Each sector can contain 11 elements selected from barriers, alarms and response force. The input of these elements is expedited by the use of a central database, so a practitioner only needs to select the desired element from the existing options.

Once input, elements may have their parameters adjusted to better capture their state. As an example, alarm systems feature a detection parameter that is based on the assumption they are perfectly situated and in perfect working order. A reduction parameter can be applied to help capture variance from the ideal.

Once the elements have been established, possible attack styles can be input. The attack styles are considered as a list of predefined possibilities which add up to 100%. The attacks are modelled on a per sector basis to allow for perimeter peculiarities, such as sections which are easier to climb or prone to attack from more highly motivated inmates.

To capture variances in security between day and night it was necessary to repeat the process and create a second solution. These variances result due to staff reduction and inmate confinement at night, altering the areas open to attack and the staff on hand to respond. Tarr and Peaty (1995) suggest that multiple solutions can be created to capture any significant variances in arrangement allowing for a more thorough exploration of the problem space.

Once the information has been input, CLASP is able to perform its analysis. CLASP produces a report that includes information on the Probability of Attack (PA)[4], Probability of Detection (PD), Probability of Intervention (PI) and a breakdown of attack style success. The information provided by CLASP helps to highlight areas of vulnerability and risk.

Using the feedback from CLASP the researchers then examined various changes to the perimeter of the sectors at risk, with details provided on the process undertaken to improve sector four (See Figure 2.8). They discuss the application of CLASP to examine the cost and benefit of various scenarios including additional barriers, alarms and response forces. CLASP allowed them to not only examine the impact on PD and PI of changes but also cost of implementation and maintenance.

As covered previously, CLASP supports the ranking of solutions based on defined goals. Using this ranking system they were able to compare six possible solutions to the existing perimeter, allowing a user to make an informed choice on which solution is most appropriate. They emphasise that the final selection is left to the user to allow for outside considerations such as planning permission or procurement difficulties.

In conclusion, they discuss that CLASP is best utilised to examine specific facilities as opposed to more general policy. The system can be used to examine more conventional designs, providing robust analysis and confidence. Alternatively it can be used to model more revolutionary designs and compare their performance with alternatives to select the best solution.

---

[4] From (Tarr and Peaty, 1995, p314); “The PA is based on the percentage of potential escapers that will use one of the seven feasible attack styles at a particular sector”. This is a user input value and is not generated by CLASP itself.

The CLASP system is an excellent example of security simulation. They discuss and show the versatility of computational analysis of security while highlighting some of its weaknesses and possible pitfalls. Their work proved very instructive to our own and provided confidence that computer aided security analysis can provide a powerful toolset for exploring a solution space.

2.2.2 Security Metrics

In Garcia (2001) we see another example of computer assisted analysis. The Estimate of Adversary Sequence Interruption (EASI) model was developed by Sandia National Laboratories to assist in security modelling. In particular, the EASI model computes the probability of interruption along a set path for a set of attack and defence variables.

By modelling detection, delay, responses and communication the EASI model generates the Probability of Interruption or P_I for a given path. While the single path analysis may prove limiting compared to a more versatile system, EASI provides a good means for quantitative analysis for the user. The output is given by the formula:

$$P_I = P_C * P_D$$

With P_C representing the Probability of Guard Communication and P_D representing the Probability of Detection. This relies upon much the same principles as the three Ds we introduced earlier; Delay, Detect and Detain. In order for an intruder to be interrupted, they must be detected with sufficient delay left before they can obtain their goal that a response force can move in and detain them.

The Response Force Time (RFT) used within EASI includes the communication time required by the guards to convey the alarm to the response force. Sandia National Laboratories has found that the chance of a successful communication to the response force increases with time and that most system operate with a P_C of 0.95, which they suggest be used as a working value unless an accurate value has been obtained for a given facility. The probability of detection itself can be broken down into the following formula:

$$P_D = P_S * P_T * P_A$$

Where P_S is the probability of abnormal behaviour being sensed, P_T is the probability of an abnormal alarm signal being transmitted and P_A is the probability of the signal being accurately assessed. P_D must be considered for each sensor along a path as an attacker is likely to encounter several in most facilities. This raises the issue of Critical Detection Points (CDP), the point along a path at which the delay value exceeds the RFT.

EASI cannot give an exact CDP value because it is based around average times and performance. It is important when considering the use of the EASI model to acknowledge that delay values will vary not just on the fitness and preparedness of an attacker but also within a given run. These variances compound across the length of a path to make exact times impossible to discern, so when considering the values used within EASI it is important to keep in mind their averaged nature and how standard deviation works.

To allow for this, when dealing with CDP they use a slightly less strict definition than what otherwise might be applied. Their experience has shown that effective systems can be designed in the face of a fuzzy CDP based on mean values, with the process otherwise much the same as when dealing with a precise CDP. The designer takes note of the CDP and attempts to increase the detection before that point and the delay after it, maximising the chance of response while trying to allow for the variance.

EASI is based upon a spreadsheet, making it a relatively easy system for a practitioner to use. Provided performance data exists for the elements making up the Physical Protection System (PPS), it can present a simple method for performing quantitative analysis on the facility. The single path modelling and need for accurate performance data present possible points of concern, increasing the burden on the practitioner to model all exploitable paths and properly acquire and understand any data for use.

The EASI model described is an excellent example of using computational assistance to perform vulnerability assessment. To our mind, the single path mapping and reliance on practitioner knowledge reduce the power of the toolset somewhat. But the EASI model is a tried and tested system and certainly an excellent reference for our own work.

In their book, Garcia (2005) provides details and discussion on the performance of vulnerability assessment against facilities. It provides a basic introduction to risk assessment and management as well as the concepts of physical protection. The book goes on to detail how to perform vulnerability assessment, beginning with planning the project and assembling a team.

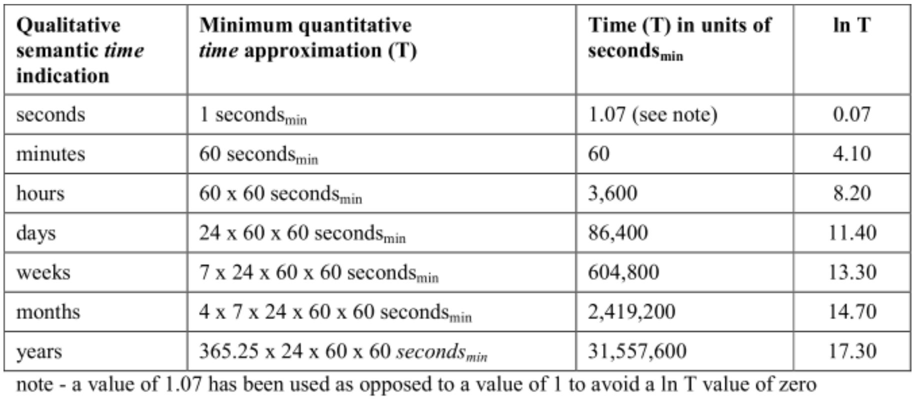
It describes the data collection and analysis process, including tips on performing simulations to gather data. It then details the reporting process, to assist in the use of the results gathered. Finally, the book contains several appendices giving examples of worksheets for project management, data collection and example delay times for a selection of barriers.

The information contained within the book assisted greatly in our understanding of security concepts. Coming from a computing background, the book provides a great deal of detail to consider that a reader might otherwise neglect. We also found the appendix on representative delays a very useful reference, helping to expand our available reference points for delay data and supplement the knowledge we obtained from Alach (2007).

In his Masters thesis, Alach (2007) presented his work on security materials. His work used a questionnaire to garner responses from a range of security personnel, from students to professionals, to estimate the delay quality of various barriers. Alach then performed quantitative analysis on those responses to create knowledge matrices which capture the domain knowledge of those involved.

Alach discussed the difficulty in choosing between qualitative or quantitative assessment within security and how this lead to the adoption of a process that attempts to marry the two. Qualitative assessment should provide more realism in the results but requires a less restrictive format among a relatively small group of responders. Quantitative assessment will provide, in general, greater precision but requires a larger number of responders.

To map the quasi-qualitative delay values provided by security practitioners, Alach used a logarithmic function. This mapped the fuzzy time estimates provided by practitioners in response to the questionnaire to a numerical value. The mapping of these values can be seen in Figure 2.9, with the T value from the final column being used to express material delay in the rest of their work.



Qualitative semantic *time* indication	**Minimum quantitative *time* approximation (T)**	**Time (T) in units of seconds$_{\text{min}}$**	**ln T**
seconds	1 seconds$_{\text{min}}$	1.07 (see note)	0.07
minutes	60 seconds$_{\text{min}}$	60	4.10
hours	60 x 60 seconds$_{\text{min}}$	3,600	8.20
days	24 x 60 x 60 seconds$_{\text{min}}$	86,400	11.40
weeks	7 x 24 x 60 x 60 seconds$_{\text{min}}$	604,800	13.30
months	4 x 7 x 24 x 60 x 60 seconds$_{\text{min}}$	2,419,200	14.70
years	365.25 x 24 x 60 x 60 *seconds$_{min}$*	31,557,600	17.30

note - a value of 1.07 has been used as opposed to a value of 1 to avoid a ln T value of zero

Figure 2.9: Mapping approximate times to appropriate values from Alach (2007)

When asking respondents how long a particular attack would take against a given element, Alach allowed for them to select the qualitative time indications in the first column.

He then converted these values into a numerical second value, expressed in the second and third columns. The final column provides the converted T value after applying a logarithmic function.

The number of potential combinations of defences and attacks is self evidently vast, so rather than attempt to map all elements, Alach worked to create a set of key data points. Alach proposed that these data points can be used to infer other values not directly captured. An example dataset in which six attack elements were each compared to six barrier elements can be seen in Figure 2.10.

In Figure 2.10 we can see Element 1 lists the attack method while Element 2 lists the material it will be utilised against. The mean value was derived from the estimated time responses from Alach's 26 respondents. The values provided are converted using the information from Figure 2.9.

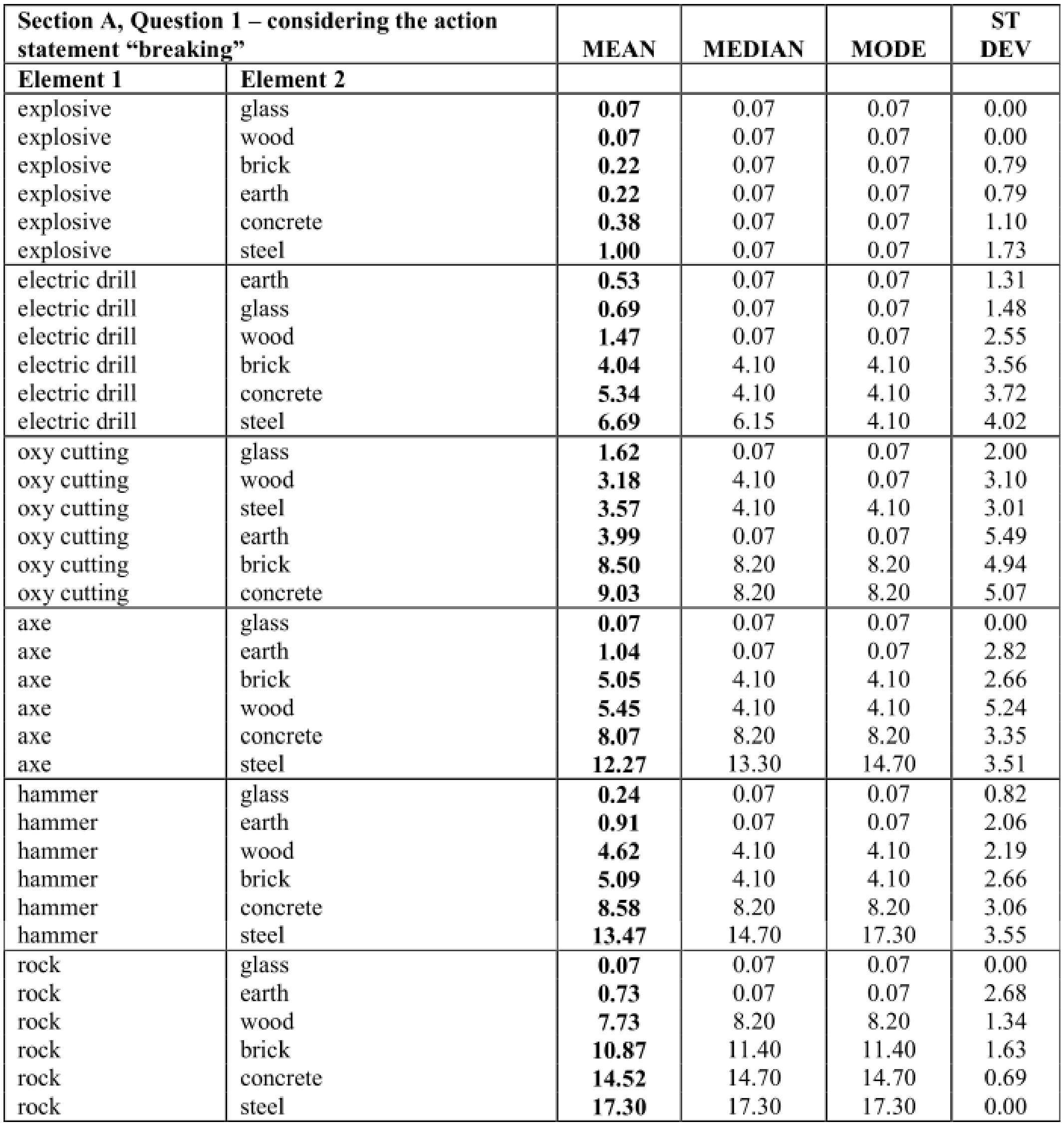
Within his thesis Alach provides several appendices, mapping various attacks to appropriate delays based on responders impressions. When viewing these appendices the use of the time mapping must be kept in mind. However, with the key from Figure 2.9 in mind, a reader is provided with a wealth of data suitable for use as working values at the very least.

The appendices he provides include data values for extended attacks and delays, such as data on time to climb barriers of various heights and materials. Data is also provided on expected likelihood of an attacker of various degrees of preparedness and motivation to go after a target of a given value. A great deal of data is presented by Alach and it can provide a basis for many aspects of security simulation or analysis.

Overall we found the work by Alach to be an excellent source of base values for security modelling. We have used his findings as the basis for our material delay values, which we will discuss in greater detail in Chapter 4. His background and research may also provide a good foundation for those new to the field of security in some of the aspects of thinking required for vulnerability assessment.

2.3 Summary

In this chapter we have provided a summary of the papers that helped provide a foundation for our own work. We have shown work in the fields of Building Information Models, Graph



Section A, Question 1 – considering the action statement "breaking"		**MEAN**	**MEDIAN**	**MODE**	**ST DEV**
Element 1	**Element 2**				
explosive	glass	**0.07**	0.07	0.07	0.00
explosive	wood	**0.07**	0.07	0.07	0.00
explosive	brick	**0.22**	0.07	0.07	0.79
explosive	earth	**0.22**	0.07	0.07	0.79
explosive	concrete	**0.38**	0.07	0.07	1.10
explosive	steel	**1.00**	0.07	0.07	1.73
electric drill	earth	**0.53**	0.07	0.07	1.31
electric drill	glass	**0.69**	0.07	0.07	1.48
electric drill	wood	**1.47**	0.07	0.07	2.55
electric drill	brick	**4.04**	4.10	4.10	3.56
electric drill	concrete	**5.34**	4.10	4.10	3.72
electric drill	steel	**6.69**	6.15	4.10	4.02
oxy cutting	glass	**1.62**	0.07	0.07	2.00
oxy cutting	wood	**3.18**	4.10	0.07	3.10
oxy cutting	steel	**3.57**	4.10	4.10	3.01
oxy cutting	earth	**3.99**	0.07	0.07	5.49
oxy cutting	brick	**8.50**	8.20	8.20	4.94
oxy cutting	concrete	**9.03**	8.20	8.20	5.07
axe	glass	**0.07**	0.07	0.07	0.00
axe	earth	**1.04**	0.07	0.07	2.82
axe	brick	**5.05**	4.10	4.10	2.66
axe	wood	**5.45**	4.10	4.10	5.24
axe	concrete	**8.07**	8.20	8.20	3.35
axe	steel	**12.27**	13.30	14.70	3.51
hammer	glass	**0.24**	0.07	0.07	0.82
hammer	earth	**0.91**	0.07	0.07	2.06
hammer	wood	**4.62**	4.10	4.10	2.19
hammer	brick	**5.09**	4.10	4.10	2.66
hammer	concrete	**8.58**	8.20	8.20	3.06
hammer	steel	**13.47**	14.70	17.30	3.55
rock	glass	**0.07**	0.07	0.07	0.00
rock	earth	**0.73**	0.07	0.07	2.68
rock	wood	**7.73**	8.20	8.20	1.34
rock	brick	**10.87**	11.40	11.40	1.63
rock	concrete	**14.52**	14.70	14.70	0.69
rock	steel	**17.30**	17.30	17.30	0.00

Figure 2.10: A break down of the six barrier elements vs the six attack elements from Alach (2007)

Theory and Security Simulation that are relevant to the expansion or understanding of those fields. The preceding review informs us of the complexity of the elements involved and to understand the work we will discuss ahead.

In the next chapter we will examine Building Information Models with reference to the tools available for their manipulation, why we selected the toolset we did and the methods we used to extract necessary information from the data stratum within Building Information Models.

Chapter 3

Building Information Models and Graphing

3.1 Introduction

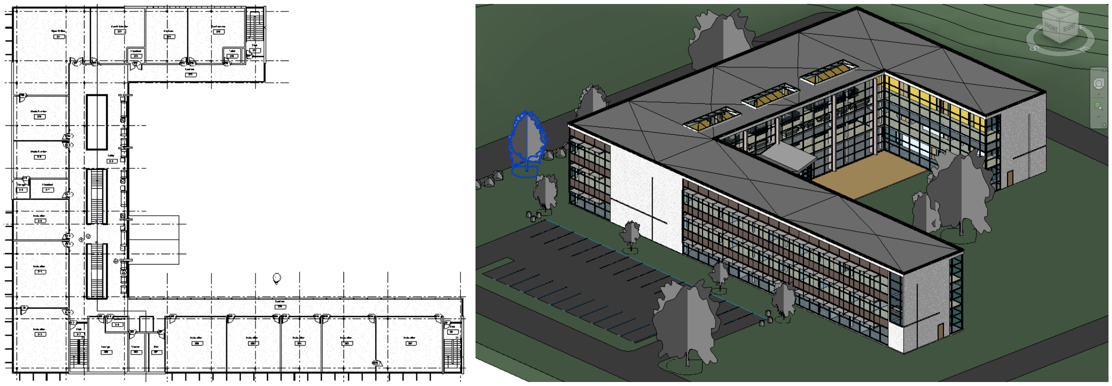


Figure 3.1: An example Building Information Model shown as 2D floorplan and its 3D representation

Building Information Models (BIMs) are an evolving data standard used to represent not just buildings but entire facilities and the assets within them. It is easiest to think of a BIM as a 3D Computer Aided Design (CAD) model with an added database layer, allowing components within the model to reference data relevant to the object. For instance, in a traditional 3D CAD a designer could portray a model of a car engine, but the model would only contain geometric information, while a BIM model of the same could encapsulate details of temperature and pressure thresholds for the pipes and valves involved.

This ability of a BIM to capture pertinent details and roll them into the model, allowing a user to interrogate the model for specific information is the source of BIMs great versatility and potential. This addition of a relational database allows BIM to be used not just during design and construction of an asset, but throughout its life cycle to provide further value add to the stakeholders. For instance, given the information that can be associated with objects, should a pipe burst the BIM can be interrogated to recover stored information such as who installed it, who last serviced it and what part number it is, making it easier for the asset manager to appropriately act.

The information layers in BIM also support greater intelligence in the software, allowing for time saving measures such as automated conflict detection, cost estimation and more as the technology develops. This power is driving uptake of BIM in the construction, design and asset management industries and is what lead to our interest in it. By exploiting the information layer it is possible for us to perform complex analysis without recourse to manually entering data.

As BIM use becomes more widespread, it will allow for our work to become another value add, easily integrated into a project and run with a minimum of extra data input. The main caveats to this will be our systems ability to harvest information from the BIM and correctly map it to a useful structure, but if these can be addressed then BIM will provide a strong foundation for our work. Within this chapter we shall detail the work we have undertaken with BIM and the results thus far produced.

It is worth noting that, at present, not all information falls within the purview of BIMs. As BIMs tend to map facilities, large scale distributed systems such as sewage and roads are often recorded separately in Geographic Information Systems (GIS). Some data has simply yet to be ported across to BIM from CAD or other earlier representations. Elements such as piping which are considered relevant to a facility may be modelled within the BIM, but typically only to the property boundary or within the facility footprint.

Solutions to help bridge this gap in coverage exist. PenBay[1] have a suite of products providing GIS services, with support for the exchange of data between BIM, CAD and several other data formats. We do not currently have a way to use any information contained within the GIS layer in our modelling, but the incorporation of other data, if necessary, should be considered in the future.

In the first section of this chapter, we shall provide discussion of many of the available BIM toolsets. We will cover the capabilities of the toolsets available and the criteria upon which we selected a toolset to work with. We do not claim to represent all toolsets for working with BIM but will examine several of the available candidates.

In the second section we will present our method for abstracting information from BIM and forming it into a graph for analysis. We will present our method as both an abstract model as well as more specific details of its implementation within our chosen BIM toolset.

---

[1] http://www.penbaysolutions.com/

In the final section of this chapter, we will present and assess the results of our work with BIMs. We will provide examples of the graphs created with reference to the models they were created from.

3.2 Building Information Model toolsets

There is a variety of toolsets available for interacting with BIMs. In this section we will discuss a selection of those available. The toolsets available can be broadly divided into open source and commercial, and we will discuss them in the following sections.

3.2.1 Open Source toolsets

Open Source software is typified by being free to acquire and distribute with source code, allowing users to extend the software and further its distribution. Typically this distribution is subject to the user allowing future users to similarly develop and distribute the software freely. Given this paradigm, some large usage open source projects have thousands of users also acting as developers.

The open source paradigm clearly allows for a large developer base, but due to the free distribution requirements typically enforced by open source licensing, most of these developers will only work on the project in their free time. This can result in slow advancement or projects becoming abandoned when they have only a small developer and user base, but the free distribution of well supported projects can make open source a very useful basis for a project.

An aspect common across the open source BIM projects is the use of Industry Foundation Classes (IFC). IFC is developed by BuildingSMART [2], an industry lead consortium dedicated to maintaining IFC as a neutral product model. As such, IFC is a non-proprietary format that can, in principle, be freely exchanged between commercial and open source products.

When we conducted research into available platforms at the beginning of our work, only two were identified; OpenIFC [3] and IFCOpenShell [4]. We used several criteria when

---

[2] http://www.buildingsmart-tech.org/

[3] http://www.openifctools.eu/

[4] http://ifcopenshell.org/

examining the toolsets including; Extensibility, Support, Stability and Functionality. As we will cover below, we found the examined systems inadequate for our needs and were forced to adopt a commercial platform.

Because of the differences in various APIs and platform features, we became locked into our selected platform a few months into development as the amount of rework required for platform change became un-viable. IFCOpenShell has matured since our platform lock-in date and new open source products, such as the IFCPlusPlus [5] and xBIM [6], have become available. Given these developments others should research the available platforms to see if they match their criteria.

One of the first toolsets we considered was OpenIFC. The toolset was appealing due to its Java based development, allowing for cross platform use and development. Unfortunately the project was discontinued as of 2010, sometime before our research began and as such the toolset was discounted due to lack of support.

IFCOpenShell was another toolset we examined for use. It provides open source support for reading and writing IFC files, but at this stage the support for geometry is somewhat limited. While the toolset is still undergoing development, at the time of our evaluation only version 0.2 was available.

Progress for this toolset has been slow and as of 11th of February 2013 they have only released version 0.3, with a release candidate for 0.4 available as of 6th of January 2014. They have plugins for 3D Studio Max[7] and Blender[8] available, allowing users to create IFC files from 3D models within those programs, which adds to its versatility and accessibility. Blender is also available as an open source project allowing for an essentially free BIM experience. However, the slow development of the IFC Open Shell toolset and limited geometry support lead to us discounting it as an option.

We also studied the IFCPlusPlus toolset. This toolset, as may be evident from its name, is built upon the C++ programming language. It is distributed for Windows but the source code is made available, allowing users of other platforms to potentially compile the code for themselves. The toolset provides a competent visualisation suite, including various display modes such as wire frame. It can produce structural analysis and provides a GUI to allow browsing of the IFC model in 3D.

---

[5] http://www.ifcplusplus.com/

[6] http://www.openbim.org/

[7] http://www.autodesk.com.au/products/autodesk-3ds-max/overview

[8] http://www.Blender.org/

The toolset seems quite promising and might provide a good open source basis for any future project iterations. Its open source nature and included toolset should provide a strong foundation for a project utilising IFC and is worth further investigation. Unfortunately the system only became available in early 2013, considerably later than necessary for use in this research.

The xBIM toolset is another promising system under ongoing development. It is the result of co-operation between the BIM Academy[9] and Northumbria University, instigated by Professor Steve Lockley. The project takes advantage of functions within the C# language to make the system more easily extensible, adding to the ease of developers improving the core functionality as required.

The core functionality allows for the access, viewing, editing and writing of IFC files. The toolset also includes a 3D visualiser, allowing for IFC files to be viewed in full 3D from varying angles and details from the information layer to be assessed. It supports functionality such as transparency, providing a reasonably powerful visualisation system.

Of the open source tools we have examined, xBIM seems like the best basis for development. The combination of IFC tools and visualiser created in open source should allow for reasonably easy extension. It is also under ongoing development from a dedicated team who seem to be making good progress and provide good support.

It is unfortunate we did not become aware of xBIM earlier in our systems development. From a comparison of the project and our development timeline, we believe it did not become available until several months after we had conducted our toolset search and adopted a platform. As BIM usage increases it is likely more projects will arise and it is worthwhile for developers to assess and reassess their options until they reach a lock-in point.

We did also consider development of our own toolset to implement IFC reading. It was decided that the development of such a toolset would consume too much effort that might detract from our research aims. Instead we looked towards commercial toolsets for a more quickly implementable solution, as we will cover below.

In general, when we undertook our primary research into open source solutions, the tools seemed to be often incomplete or poorly supported. For our own project with limited time, this presented a barrier to our uptake of any of the tools we reviewed. This is unfortunate because use of open source tools can often lead to feedback and development that improve

---

[9] http://collab.northumbria.ac.uk/bim2/

the tools themselves as well as reduced licensing costs.

The improvements in the available tools we have discovered since our initial platform research bodes well for other developers looking to utilise open source. We look forward to seeing what the community produces in the future.

3.2.2 Commercial toolsets

There are several commercial toolsets available, with most major vendors of CAD authoring tools also having a BIM authoring option. These packages are typically well supported, highly developed and widely used within industry. Their downside comes from the high cost to license such professional level tools and their typically closed ecosystem.

We primarily examined the BIM offerings by Bentley® and Autodesk®, two of the main distributors of CAD and BIM authoring tools. Both companies offer a wide variety of software packages within the realm of design, providing a digital ecosystem within which a user may work. Among the products on offer, a small subset are intended for use with BIM, with the available products varying in their intended roles or scope within a project.

While both companies support IFC to some degree, they also make use of proprietary file formats and typically use internal synergies between their products to encourage users to make use of their product ecosystem.

Bentley® offer a large selection of tools for working within architectural design. Many of these tools act as extensions for Bentley Microstation and their BIM toolset is titled Bentley® Architecture. This toolset offers a large feature set useful to designers looking to create facilities[10].

Bentley® Architecture supports high level creation and editing of BIM in various file formats including Design (.DGN), their proprietary format. It can generate renders and animations of facilities to help visualise and demonstrate functionality to users. It supports full 2D and 3D design and editing for facility design. In brief, it provides most of the tools needed for creating, editing or visualising BIMs.

Bentley® also provide a series of APIs (Application Programming Interfaces) and SDKs (System Development Kits) for creating extensions for Microstation. The software suite is

---

[10] http://www.bentley.com/en-AU/Products/Bentley+Architecture/

well supported and the APIs appear to be well supported, opening up good potential for extensibility. This all made Bentley® Architecture a promising option for development.

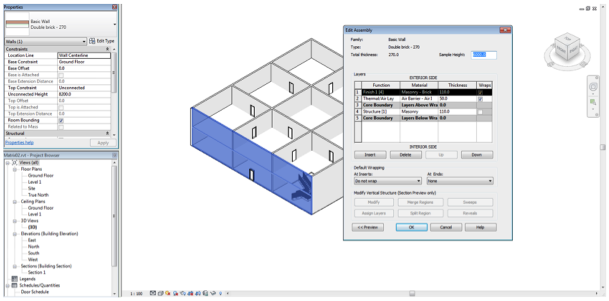


Figure 3.2: A screengrab from within Revit®

Autodesk® Revit®[11] is the BIM solution offered by Autodesk®. It offers a large suite of features, comparable to those offered by Bentley® Architecture. The software has an accessible interface that offers 2D and 3D modelling with visualisation options.

We would note that while Revit® claims support for IFC import and export, our own experiences have been mixed. We have noticed when exporting and then re-importing an IFC file into Revit®, some planes and objects tend to vanish or move within the model. Informal discussion with other users and searches on the internet[12] suggest this is a commonly encountered issue with Revit® and IFC. This is a limitation that needs to be addressed to increase the functionality of Revit®.

Revit® do provide an API allowing developers to create add-ins that run within the environment against loaded models. Their developer and student community are both active and support is available for new developers. They also provide a student access program, allowing university students to register and access their software for free. Between the toolset, developer support and student access program Revit® provides a compelling option.

While the software available from both vendors is fairly equivalent, we ended up selecting Autodesk® Revit®. Their student licensing model was more transparent and made the software easy to acquire. We cannot recommend one suite over the other based purely on features and others should examine the software for themselves with reference to their institution's licensing.

It is also worth mentioning Solibri[13]. They offer a small suite of three products with

---

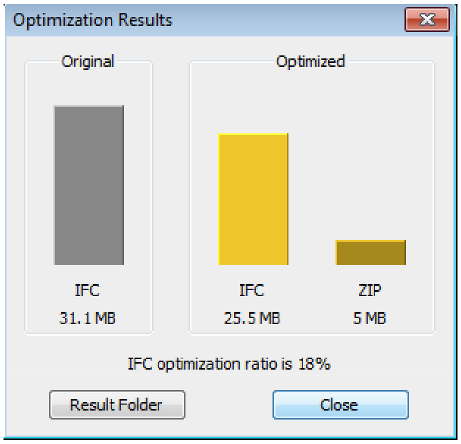
[11] http://www.autodesk.com/products/autodesk-revit-family/overview

[12] http://deliverysimulation.com/revit-the-problem-with-ifc-exports/

[13] http://www.solibri.com/

both the Solibri Viewer and Solibri Optimiser made available for free with registration. The Solibri Model Checker is available under license and can be used not only to check a building for potential conflicts and weaknesses but also for compliance with building codes and best practice.

Due to its licensing, we have no first hand experience of the Solibri Model Checker but have made use of the Solibri Optimiser. The Optimiser takes in an IFC file and performs analysis for redundancy within the file structure, removing duplicates to reduce the file size. They claim compression down to 6% of original size and a reduction in subsequent load time down to 30% of original. From our own testing we are yet to see such a spectacular result, but have recorded reasonable reductions in the size of test files, as can be seen in Figure 3.3.



IFC optimization ratio is 18%

Figure 3.3: A sample file optimisation performed by Solibri IFC Optimizer

3.3 Method For Mapping Information Within BIMs

In this section, we shall describe our method for extracting information from the BIM. Through necessity, where code is shown, it will deal with the particulars of working with the Revit® API. To assist users of other software packages who might wish to replicate our results, we will concentrate on describing our process and reason.

After extraction, the information will be organised into a graph format. A graph representation was chosen due to the established methods for searching and manipulating them within computer science. Use of these established methods, such as Dijkstra's algorithm, allow us to more quickly and confidently explore the problem space.

3.3.1 Graph

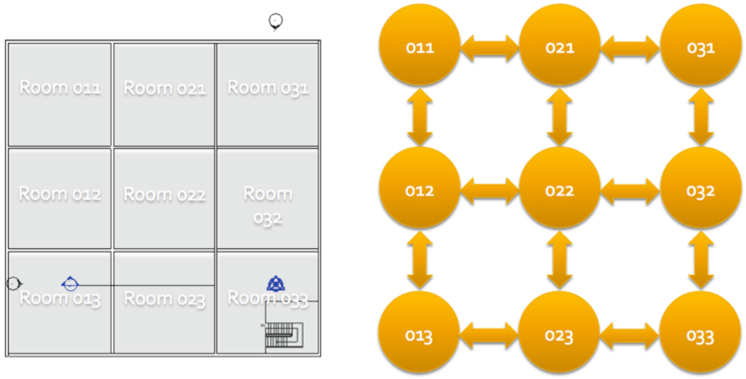


Figure 3.4: A simple floor layout beside a simplified graph representation of the same.

A graph is made up of nodes and edges. Edges are typically assigned cost values and the cost to move from one node to another can be calculated by using the edges and nodes between them to find a path. Nodes thus make a natural analogy for destinations.

When considering the problem of graphing a facility, we use objects to represent the edges and the nodes. Within those objects we embed links to allow us to navigate the graph. For graphing a facility, edges can be anything that connect from one area to another, leading to nodes typically having multiple edges between them. The graph needs to capture the physical description of the facility, with edges matching all foreseeable paths through the facility.

So when mapping a facility an edge can be a physical boundary, such as a wall, or a spatial boundary, such as distance, or even a conceptual boundary. Conceptual boundary edges may not have any cost but are important to include to accurately capture a facility layout and make it easier for a user to understand how a graph was traversed. The cost to travel within a room is not captured by the above mappings.

We map areas to nodes, so a node will typically represent something like a room. But if a room is a hallway, the cost to travel from one end to the other may be non-trivial. We could perhaps break nodes down to add edges within a node, but our graph would quickly become very complex and confusing for a user.

A possible solution would be to introduce a hierarchy, where a area could be examined and its node broken down based on its dimensions. Then when traversing the graph if a "parent" node is encountered, the extra cost of getting from one edge to another could be calculated by traversing the child graph. This adds complexity to the graph traversal but is a reasonable solution to the problem.

There are also other possibilities, such as consulting the area diagram to generate an appropriate additional cost depending on which edges are being accessed and their physical locations. The complexity of any of the solutions increases as we try to capture more accurate costs and so we decided that solving the problem fell outside the scope of our current research. It is something we would like to see solved in the future and discuss some of the possibilities for doing so under future work.

3.3.2 Rooms, Areas, Walls and Conceptual Edges

The first task for generating a graph of the facility is what we map to the nodes. As the central points that must be navigated between, we map the nodes against areas and rooms. This allows us to capture the layout of a facility and the way rooms and areas might interact.

As an example of an area you might think of a room, yard or a car park surrounding a facility. Car parks and yards are not always modelled as areas, but if they are cordoned off in some manner such as by a surrounding fence, it can be necessary to capture them to accurately model the facility. By mapping these areas it makes it possible to simulate the possible paths in and out, increasing the areas that can be examined for change.

The mapping of these rooms and areas through automated means is a complicated task. The reason being the areas must be clearly defined and within buildings these areas may not be. For instance, if we looked to detect rooms by simply finding areas encapsulated by walls, we run into the problem that a single wall may border many rooms.

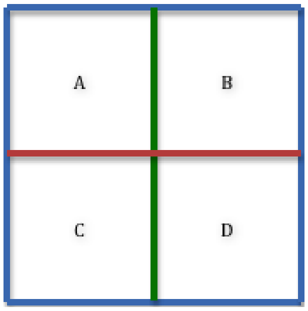


Figure 3.5: A simple floor layout with walls coloured for identification

Using the floor layout in Figure 3.5 as an example, we can see the complexity. The floor layout provides for four rooms, but if we try to determine this purely from the walls, we might select the red wall and the top or bottom blue wall, creating two long rooms. Likewise, the algorithm might find the blue walls and create a single large room. We could perhaps work around this by having the algorithm work down to the smallest room, eliminating any room containing another. This would introduce problems with accuracy and require the comparison of new rooms against all others in order to detect such a conflict, vastly increasing algorithmic and computational complexity.

Similarly, an outdoor area may not be bounded on all sides but still be required for proper facility modelling. To capture such an area through automated boundary detection is difficult but to accurately model the facility, such areas need to be allowed for. For instance a private road or car park may not be walled off but still provide relevant access paths.

It was decided that the algorithms required to work purely from wall bounding data would be either too computationally expensive and complicated or not accurate enough. While exploring solutions to the problem, we found the room label tool provided within Revit®. Using this tool, you can click inside an area and Revit® will fit the room within the available space by intelligently following curved and straight boundaries, allowing you to easily label areas with the option to manually alter boundaries when required.

Using the room tool we place a labelled area within each room and hallway within a model. We also use the room tool to label any areas outside that we wish to be modelled. In the case of hallways, several room labels may be necessary to capture interconnected areas.

With room areas designated, we now face the much simpler problem of matching boundaries to the rooms they adjoin, using a brute force search method. High level pseudo-code for our method can be found in Figure 3.6 and we will now describe the method in detail. It is worth noting that within Revit®, both separation lines and walls are subsets of Spatial Element Boundaries, so we can check against Boundaries and later concern ourselves with whether the boundary is a wall or separation line.

We begin matching by iterating through each room and then iterating through all boundaries and comparing their bounding boxes to identify those that form the edges of our current room. When such a wall is found, we iterate through all other rooms to match any that have a bounding box also intersected by the wall, suggesting the two might be neighbours.

Once we have the two possible neighbours, we run checks to make sure they can be neighbours. We check to see if they share the wall in question. We establish that they

```
FOR EACH room
  FOR EACH boundary of the room
    FOR EACH segment of the boundary
      FOR EACH other room
        FOR EACH compareboundary of the otherroom
          FOR EACH comparesegment of the compareboundary
            IF segment = comparesegment AND
            IF room.elevation = compareroom.elevation AND
            IF BarrierBetween(room, compareroom) THEN
              Create the barrier as an edge between the two nodes
```

Figure 3.6: Psuedocode to compare room proximity via wall barriers

occur on the same elevation. We confirm they should lie on opposite sides of the wall. If all of these are found to be true, we accept them as neighbours and create the edge between the relevant nodes on the graph.

3.3.2.1 Corners, Bounding Boxes and BarrierBetween()

Within Revit®, we came across a slight problem with matching. In the case of rooms which share a corner, we found that our algorithm was often matching these rooms as neighbours. For instance, within Figure 3.5 rooms A and D would be connected by virtue of the corner. From experimentation, this would appear to be due to the bounding boxes on the walls and rooms typically being just slightly larger than the objects themselves, producing overlap.

To correct this we created the BarrierBetween() method. The method takes in as its arguments three objects; roomA, barrier, roomB. It then uses the bounding data from these objects to try and determine adjacency.

It first works out if the X midpoint for both rooms fall within the X range of the barrier. If they do, we then check to see if their ranges overlap, which we do by checking to see if the X midpoint of roomB falls within the X range of roomA. If both of these tests are passed, we know the rooms and barrier overlap vertically but then need to see if the barrier lies between the two rooms.

We do this by use of the Y values of the barrier and rooms. We check if the Y values for the midpoints of roomA and roomB would place them on opposing sides of the barrier. If all the above conditions are true, the barrier lies between the two rooms.

If both conditions are not met, we also repeat the above tests with the X and Y checks reversed. While this method has proven effective for most room layouts, it has difficulty with some unusual room layouts that involve curved or angled barriers. Revision to allow for more reliable detection of neighbours with varied barrier orientation should be undertaken in future work.

As an example of a normal comparison, look at Figure 3.7. Room A and Room B would be matched, as their midpoints both fall within the range of Barrier A, their midpoints clearly fall within each others X range and their Y values would put them on opposing sides of the barrier. Room A and Room C will not be matched because while their X midpoints fall within Barrier A's X range, Room C's midpoint is not within Room A's X range.

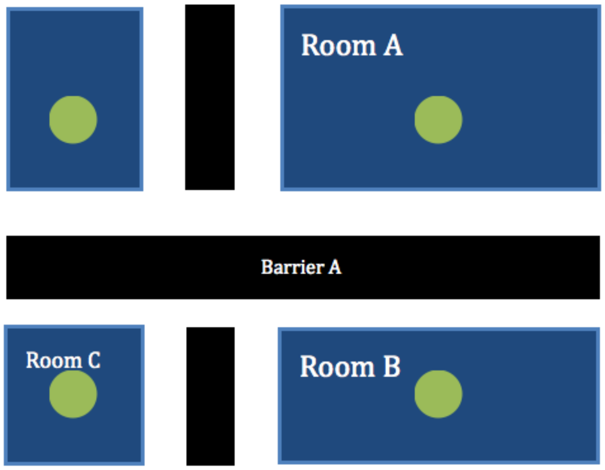


Figure 3.7: Black lines represent barrier, blue blocks areas and green dots the mid points.

Once we have confirmed the barrier separates the two rooms, we use element ids to look up the rooms and see if we have existing objects representing them. If we do, we load those, if not we create new room objects and load them with data from the room elements we have. We then check the barrier elements category to see if it is a "<Room Separation>" and generate the relevant conceptual barrier object. If it is not, we assume its a wall and create the relevant wall object.

Once we have all three objects, we then link them to each other, giving us the edge between two nodes. This is repeated for all barriers on the room, then all other rooms until we have a near complete neighbourhood graph for a floor. In the cases where our matching cannot find another room on the other side of a barrier, we instead link to the outer zone.

The outer zone is a object we create to represent everything outside the facility. If we can not find a room on the other side of the barrier, we assume it must be an external wall. We then link the room object to the outer zone object via the barrier object.

With all this done, we consider ourselves to have a complete graph of areas with wall and spatial edges. We have also created objects to represent all rooms and areas. We next turn to the problem of mapping doors and windows.

3.3.3 Windows and Doors

The process for windows and doors is similar to the one used above for walls and separations. Unlike walls and separations, windows and doors are distinct families, so we have to perform one matching operation then the other separately. Rather than state the steps twice, we shall go through an example with doors and it can be assumed that windows may be substituted directly for doors unless noted.

Within Revit®, we perform a filter element operation to restrict down to objects within the BuiltInCategory.OST_Doors. For each element, we then perform a category name check to ensure the element is indeed a door as a precaution against unknown objects. If we confirm it is a door, we instantiate a door class object and inform it with data from the element.

The process for linking to appropriate area nodes is much simpler for doors as the door element contains FromRoom and ToRoom fields. We use these fields to retrieve the element id of the rooms, if they exist, and use the id to load any existing room class objects with those ids. Because all rooms were created in the previous section, we can trust that if a room cannot be loaded, then it does not exist or is improperly labelled. In either case, our fallback position is to assign the outer zone to the field we could not match.

Once we have established the FromRoom and ToRoom, we link the rooms to the door and the door to the rooms, allowing for the graph to be easily traversed. We do have a check to ensure that the from room and to room are not the same room, as internal doors present an issue when traversing the graph as they may result in loops. To avoid this we perform the check and ignore doors that appear to be internal to a single room.

At this point the edge has been established or dismissed and we move on to the next element. Once all doors have been completed, as indicated, we repeat the above process in a separate code loop for windows. The high level pseudo-code for this process is shown in Figure 3.8.

```
COLLECT ALL DoorElements
FOR EACH DoorElement
  CREATE DoorObject
  IF Element.FromRoom != null
    DoorObject.FromRoom = FromRoom
  ELSE
    DoorObject.FromRoom = OuterZone
  IF Element.ToRoom != null
    DoorObject.ToRoom = ToRoom
  ELSE
    DoorObject.ToRoom = OuterZone
```

Figure 3.8: Psuedocode to compare room proximity via door barriers

With walls, separators, doors and windows all added as edges, there are no other horizontal edges necessary to model. In the next subsection, we will begin to deal with vertical neighbours, starting with floor and roof edges.

3.3.4 Floors and Rooves

```
COLLECT ALL FloorElements
FOR EACH FloorElement
  FOR EACH RoomElement
    IF RoomElement.level = FloorElement.level
      IF pointInside(RoomElement.midpoint, FloorElement.BoundingBox)
        FOR EACH RoomElement
          IF RoomElement2.level = (FloorElement.level - 1)
            IF BoundingBoxOverlap(RoomElement, RoomElement2)
              CREATE FloorBarrier between RoomElement and RoomElement2
```

Figure 3.9: Psuedocode to compare room proximity via floor barriers

Floors and Ceilings, similar to walls, may cross multiple areas as can be seen in Figure 3.10. Floors are in some respect simpler to work out, as a room can be either above or below a floor. This means we need less error checking when compared to walls to avoid problems caused by things like corners.

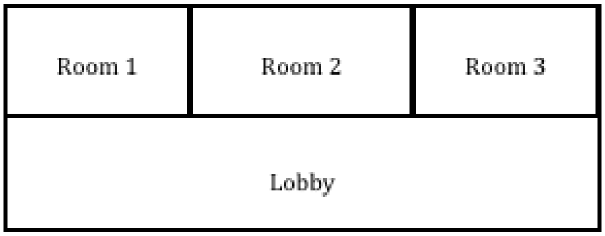


Figure 3.10: A simple two level cross section

The basic method for comparison, as described in Figure 3.9, is to select a floor and then find a room above this floor. We then search for rooms below the floor and, if they overlap the first room, we create the edge link. Unlike the edges above, we do not link floors to the outer zone.

This is primarily because of the complexity of correctly identifying the cases where this is necessary. In traditional architecture it is not too difficult and we could most likely use a simple boolean switch to track whether one or more neighbours have been found and if not, add an edge linking to the outer zone. However we considered the case of modern architecture design where a area will sometimes overhang, which should cause a link between a room above and the outer zone.

To properly check for these we would need to track the overlap of rooms below to calculate any surface of the room that does not overlap a room below. While this is certainly possible, it added a layer of complexity we did not have time to address during our current round of research. We also assumed that in the case of ground level rooms it is unrealistic to link to the outer zone through the Earth as few people are likely to access a room via tunnelling.

While sewers and subterranean tunnels might make this an exploitable edge, if they are not modelled within the BIM we cannot correctly allow for them. If they are modelled within the BIM, then they can be labelled like any other area and our existing matching algorithms should detect them and add them appropriately. It is an issue worth further consideration in the future, but our current solution would seem adequate.

3.3.5 Stairs

Stairs provided an interesting challenge compared to previous edges. They can extend across multiple rooms and multiple levels, a problem we haven't dealt with previously. They are also the first travel based edge we model.

The reason we model them but not other travel costs we've considered is they're one of the most likely ways for an entity to move from floor to floor. While we do not at present model the cost to travel from one end of a hallway to another, we could not capture the nature of moving around a building without having a measure of the cost to move from one level to another. An entity is also far more likely to take the stairs than try and move from one area to another via the floor between them.

When we first considered the problem, two methods of modelling the travel edge where apparent. We could either link each area to all other areas connected via the stairs or we could link areas sequentially. As an example, imagine a tower made up of three rooms, with room 1 on the ground floor and room 3 on the top floor. By looking at Figure 3.11 we can see a simple example of sequential linking, in red, versus all path linking, in blue.

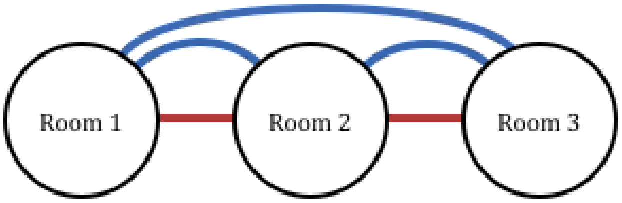


Figure 3.11: A simple one room per level example of vertical linking possibilities.

Linking along all paths creates substantially more links and the problem increases as you introduce more levels. Sequential linking provides a more elegant solution when graphing, allowing us to move from Room 1 on the first floor to Room 3 on the third via Room 2. It reduces the overall complexity of our graph without losing any fidelity in the capture of the facility.

The pseudo-code for performing this linking can be seen in Figure 3.12. We begin by creating lists of all stairs and all rooms. For each stair we then examine all rooms and record those that intersect in a separate list.

Once we have a list of rooms that intersect the staircase, we go through each room in the list and compare it to all others. Using the level information for the rooms we determine adjacency and generate links where appropriate. Once this is completed, we repeat the process for the next staircase, until all have been appropriately matched.

```
COLLECT ALL StairElements
COLLECT ALL RoomElements
FOR EACH StairElement IN StairElements
  CREATE RoomList
  FOR EACH RoomElement in RoomElements
    IF StairElement.midpoint IS INSIDE RoomElement.boundingbox
      IF RoomElement.Elevation IS INSIDE StairElelment.ZRange
        RoomList.Add(RoomElement)
  FOR EACH RoomElement in RoomList
    FOR EACH CompareRoomElement in RoomList
      IF CompareRoomElement.Level = (RoomElement.Level +/- 1)
        CREATE RoomAObject FROM RoomElement
        CREATE RoomBObject FROM CompareRoomElement
        CREATE TravelObject FROM StairElement
        LINK RoomAObject TO RoomBObject WITH TravelObject
```

Figure 3.12: Psuedocode to compare room proximity via stair barriers

3.3.6 Elevators

```
CREATE ElevatorShafts
FOR EACH ShaftElement IN Shafts
  FOR EACH ElevatorElement IN Elevators
    IF ElevatorElement.Midpoint INSIDE ShaftElement.BoundingBox
      ElevatorShafts.Add(ShaftElement)

FOR EACH ElevatorElement IN Elevators
  CREATE Match SET False
  FOR EACH ElevatorShaft IN ElevatorShafts
    IF Elevator.Midpoint INSIDE ElevatorShaft.BoundingBox
      IF Elevator.Level INSIDE ElevatorShaft.LevelRange
      OR Elevator.Level = (ElevatorShaft.LevelRange +/- 1)
        ElevatorShaft.LevelRange.Add(Elevator.Level)
        Match SET True
  IF Match = False
    ElevatorShafts.Add(ElevatorElement)

CREATE RoomElements
FOR EACH Shaft IN ElevatorShafts
  CREATE RoomsConnected
  FOR EACH RoomElement in RoomElements
    IF Shaft.midpoint INSIDE RoomElement.BoundingBox
      IF Room.Level INSIDE Shaft.LevelRange
        RoomsConnected.Add(Room)
  FOR EACH RoomElement IN RoomsConnected
    FOR EACH CompareRoomElement IN RoomsConnected
      IF RoomCompare.Level = (Room.Level +/- 1)
        CREATE RoomAObject FROM RoomElement
        CREATE RoomBObject FROM CompareRoomElement
        CREATE TravelObject FROM Shaft
        LINK RoomAObject TO RoomBObject WITH TravelObject
```

Figure 3.13: Psuedocode to compare room proximity via elevator barriers

Elevators are the other travel edge we consider. Many of the abstract graph considerations, such as sequential vs exhaustive are the same as for stairs. However, capturing their nature within a BIM is a more complicated task, as can be seen in the pseudo-code in Figure 3.13.

We found in the models we looked at, that the method for modelling elevators can vary. In some models, an elevator object was placed upon each floor where it was present. In other models, a single elevator was placed and only door models were added at other levels. Unlike stairs, the elevator element may not extend across its entire area of influence, raising the issue of how to properly associate it with relevant areas.

This raises another point of issue; Within Revit®, Elevators do not have a OST_BuiltInCategory of their own but are a type of Special Equipment. So, before we can search for an Elevator within a shaft, we must first collect all Special Equipment Elements and attempt to filter out only the elevators. Once we have done this, we can move onto the complex task of creating Elevator travel edges.

The solution we found was to look for the gaps in the building where the elevators reside. Within Revit®, a shaft can be created as a negative object that removes the floors and walls it comes in contact with. Elevator objects are then typically added within that shaft.

So to create the elevator edges, we first find all the shafts. We then go through all the elevator elements we find and attempt to match them to shafts to create Elevator Shafts. In some cases, the shaft will begin above or below the Elevator object, so we need to extend any Elevator Shaft to ensure it encompasses the area taken up by its constituent Elevators.

Once all this is done for each Elevator Shaft, we go through all Rooms looking for any that interact with it. When a Room is found we save its information to a list. Once we have checked all rooms, we then take our list of rooms and work through them to find those that are adjacent, using their elevation information.

When two rooms related to a shaft are adjacent, we attempt to connect them via an Elevator Travel Edge. We check to make sure the rooms are not already connected by an Elevator, as a second edge would be redundant under our current model, and if they are not, we create the link. Once we have done this for all related rooms, we iterate through any other shafts until all relevant Elevator Travel Edges have been added.

As our pseudo-code might indicate, this was the most difficult element to abstract into graph form. We have some concern that the lack of a OST_BuiltInCategory for elevator objects may lead to this method being unreliable on certain BIMs, if they classify the elevator object under something other than speciality equipment for instance. We have done what we can to ensure maximum flexibility. In any case, the code we developed works for a selection of test models and demonstrates the feasibility of our graph abstraction.

3.3.7 External Elevation

When generating the graph, we considered the problem of elevation. In the graphing described above, an exterior wall is linked to the outer zone. However, logically, the cost to access a wall or window on the exterior of the third floor would be greater than an equivalent barrier on the ground floor.

Adding a representative edge with cost posed a challenge to our existing graph abstraction as we would need to add some kind of area for the height edge to connect to. We did consider adding new areas to represent each level of the building, but as we will discuss in more detail in Security 4.1.1.3, we decided it would be better to simply increase the cost on the existing edges. To do this we made barriers multi layered so we could add a layer of height for each level above the ground, as shown in Figure 3.14, though this pseudo-code only deals with altering window edges.

```
CREATE LevelElevations
CREATE HeightBarrier
FOR EACH Barrier in outer.windows
  CREATE WhatLevel = findLevel(LevelElevations, Barrier.height)
  FOR 1 to WhatLevel
    Barrier.Add(HeightBarrier)
```

Figure 3.14: Psuedocode to compare room proximity via elevation barriers

This has the effect of adding an arbitrary cost to the edge for each level above the ground a barrier is. We encountered a problem when developing this method on our custom models, however, where barriers above the first floor were not having their edge cost incremented. From experimentation, it appears this is a result of our models not containing necessary information on floor elevation, used to determine the level a barrier belongs to.

We are confident this is a failing in our production of the sample models as the system has been shown to work on models produced by others. As such we have not placed a high priority on correcting it as mastering the Revit® toolset is not one of our goals and the other models tested demonstrate our method works. We may revisit this issue in the future to try and find a more robust way of determining floor that is less dependent on the elevation values, but we consider the system functional as is.

To correctly apply the height cost, we need the support of a few speciality methods and objects. The first, levelElevations, is a matched list pairing levels with their respective floor elevation. This is generated within Revit® by use of a element filter looking for

BuiltInCategory.OST_Levels.

The method findLevel() takes in the height value from a barrier and searches through the levelElevations list to find what level it resides on. It compares the height to an array of floor elevation values, matching to the highest elevation the barriers height is greater than or equal to. It then returns the matched floor level.

3.4 Results

To demonstrate the functionality of our system, we developed several BIMs internally that were designed specifically to test individual elements of our Graph conversion system. We used an iterative design approach where the BIMs grew more complex as our system was able to graph a wider variety of edges. To help demonstrate our system and show its results we created a further series of seven models.

We kept the models simple enough that the results could be easily compared to the model and manually checked. In the following sections we will present first the set-up for the graphing, including the models developed and their intention. We will then present the models and relevant excerpts of the generated data to demonstrate the effectiveness of our system.

Because our system was developed to generate graphs and perform security simulation, the output our system generates contains details for both. For the sake of making the results that relate to graphing more accessible, the results in this section will be presented with security specific details removed. Security details will be discussed specifically in Chapter 4 and unedited copies of some XML (eXtensible Markup Language) files are in the appendices A and B.

3.4.1 Testing Setup

Our software converts a BIM into an internal graph representation composed of objects to represent each node and its edges. To capture possible barrier behaviour, we have many edge class objects but only one ZoneClass object to represent areas. In combination these objects allows us to represent the graph in its entirety.

To reiterate from earlier, edge class objects will be barriers such as walls, windows, doors, stairs, elevators and conceptual. Zones class objects will be areas such as rooms and car parks. Edge class objects naturally form the edges within the graph and zone class objects provide the nodes within the graph.

Each ZoneClass object has a set of lists of each edge type. This allows any given zone to have many connecting zones via any possible edge type. Meanwhile, each edge object is set-up to accept two zones which are arbitrarily labelled as ZoneA and ZoneB. There is no preference given to ZoneA, it's simply a method for referencing one of the zones the edge connects to.

Using these objects we can capture the graph nature of all areas and edges described above. Rooms and areas become Zoneclass objects, walls becomes WallClass objects and so on. To convert this internal structure to something that can be used by other applications, our software contains a function that outputs these objects as XML.

XML is a markup language designed to make documents both human and machine readable. It relies on a bracketing system where all data is encapsulated by tags to allow for the conversion to information. A pair of tags may encapsulate data such as numbers or strings or other tags, allowing for information nesting.

For our purposes we capture the graph structure by beginning with a zones bracket. Within that bracket we then have a set of tags for each individual zone and within that zone we have some relevant information, such as name, as well as tags bracketing edge information.

For a given edge, such as a wall, the XML will contain an entry with arbitrary sequential numbering and the neighbour of the current zone, as can be seen in Figure 3.20. This is achieved by interrogating the edge, providing it with the current zone and fetching the zone from the object that isn't the current object. The neighbour is provided in the form of the neighbouring zones name, so unique zone names are required throughout the BIM to make results easily human readable.

Along with walls, the XML also contains entries for doors, windows, conceptuals, travels, roofs, floors and environments. These tags are at present included even if no relevant edges exist. These edges are not shown in the excerpts we will provide unless relevant, for the sake of keeping the examples short, but can be seen in the copies in the appendix.

To demonstrate our results we have created a series of seven models that increase in complexity along a fairly predictable path. We begin with Model 1, seen in Figure 3.15, which is simply four walls and a room, allowing us to test simple edge graphing. Model 2, on the right of the same figure, adds a door and windows, allowing us to test multiple types of edges linking to the outer zone.

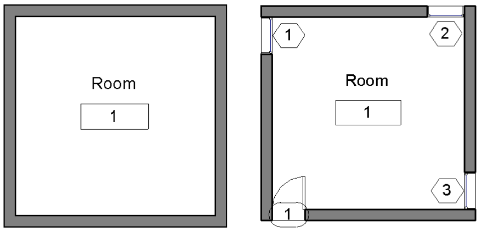


Figure 3.15: Model 1 (left) and Model 2 of our demonstration model

In Model 3, seen in Figure 3.16, we added internal walls and multiple rooms. Model 4 beside it introduces extra doors, giving us multiple edges between each zone. This allows us to test proper inter-room linking, which is an important step in mapping a larger facility.

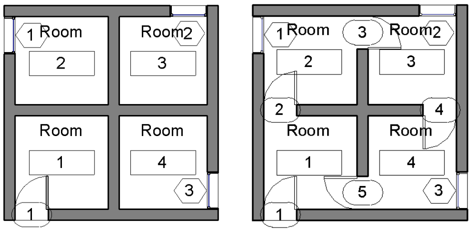


Figure 3.16: Model 3 (left) and Model 4 of our demonstration model

In Model 5 and Model 6, seen in Figure 3.17, we create a more complex series of rooms. This approach allows us to test rooms, such as the central room, are not being mistakenly linked to the outer zone. It also has interesting ramifications for shortest path, potentially creating a "chain" of rooms leading to the central room, which is most likely to be expressed in Model 7 in Figure 3.18.

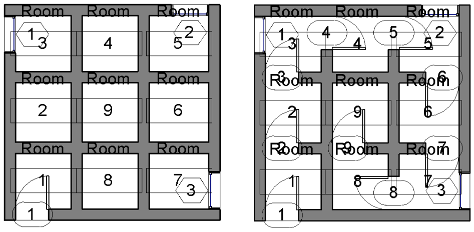


Figure 3.17: Model 5 (left) and Model 6 of our demonstration model

Some of these models are more interesting for testing the security simulation aspect, so we will only cover some of them in detail in the next section. We also have a model we refer to as the "Simple Model", which includes a set of stairs and a second floor and can be seen in Figure 3.19 on page 54.

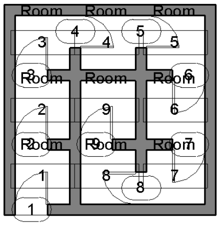


Figure 3.18: Model 7 of our demonstration model

The software itself is run as a plugin for Autodesk® Revit® 2012 Student Version. The software was written in C# under Visual Studio 2010. Both these applications were run on a virtualised copy of Windows 7 running ontop of an Apple iMac, featuring a 2.5Ghz dual core intel i5 with 16GB of Ram, with half of these resources available to the virtualised copy of Windows.

The software runtime varies depending on the complexity of the BIM. With minimal effort put into algorithm or code optimisation, the software can currently take up to several minutes for a more complex model. We would expect this runtime to reduce considerably if run on a native install of Windows running on top of modern hardware, or if more time were spent on code optimisation.

3.4.2 Testing Results

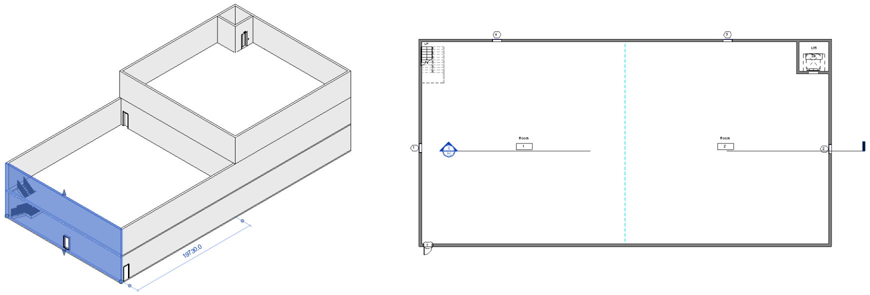
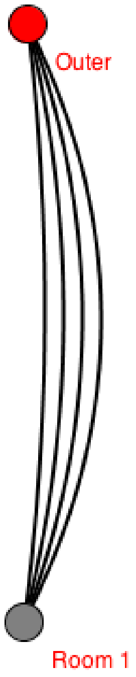


Figure 3.19: The Simple Model: A three storey model with stairs and an elevator beside its first story floorplan. A floor by floor breakdown is available in the appendices

To begin testing the graphing, we started with the simplest structure: A single room with 4 walls. This model can be seen on the left of Figure 3.15 and the edited XML excerpt can be seen in Figure 3.20, with the full XML available in Appendix A. For such a model, we would expect a fairly simple graph involving a single central node with four edges linking

<table>
<tr><th>vlstate.xml</th><th>Graph representation</th></tr>
<tr><td>

```
<zones>
  <zone0>
    <name>Room 1</name>
    <previous>Outer</previous>
    <walls>
      <wall0>
        <neighbour>Outer</neighbour>
      </wall0>
      <wall1>
        <neighbour>Outer</neighbour>
      </wall1>
      <wall2>
        <neighbour>Outer</neighbour>
      </wall2>
      <wall3>
        <neighbour>Outer</neighbour>
      </wall3>
    </walls>
  </zone0>
</zones>
```

</td><td>


</td></tr>
</table>

Figure 3.20: An edited excerpt of the XML representation of the Version 1 Graph with a visual graph representation on the right

it to the outer zone node.

Examining the XML code in Figure 3.20 we can see that the software generates and outputs the appropriate linking. We have only a single zone, zone0 which represents Room 1. Within that zone we have four wall type edges as we would expect, all of them linking to the outer zone.

On the right hand side of Figure 3.20 we present a visual representation of the same information. The red circle represents the outer zone node, the grey represents the Room 1 node. Black lines represent wall edges and as we can see the visual representation has the four we would expect.

If we look to Version 2, we would expect the same four wall links and four new links; three window type edges and one door type. The full XML can again be found in the appendices, but if we look to the graph representation and XML excerpt in Figure 3.21, we can see that the software generates and outputs all the expected edges.

v2state.xml

```
...
<doors>
  <door0>
    <neighbour>Outer</neighbour>
  </door0>
</doors>
<windows>
  <window0>
    <neighbour>Outer</neighbour>
  </window0>
  <window1>
    <neighbour>Outer</neighbour>
  </window1>
  <window2>
    <neighbour>Outer</neighbour>
  </window2>
</windows>
...
```

Graph Representation

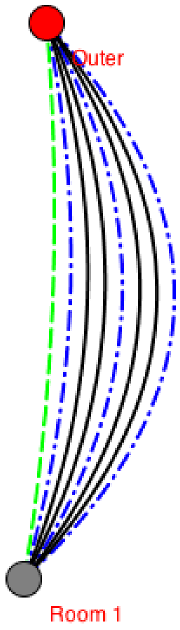


Figure 3.21: An edited excerpt of the XML representation of the Version 2 Graph

The blue dash-dot style lines indicate window edges and the green dashed line indicates the door edge. The black solid edges continue to indicate walls, as they did in the previous representation and will in all further graph representations we present in this chapter. While this visual representation can be understood more immediately than the XML excerpt, we can see with this very simple BIM that the edge complexity increases quickly.

The above demonstrates that our software works to generate the expected graphs given the two simplest scenarios. With version 3 and 4, we would similarly expect to see four wall edges per zone node, as we saw for Room 1 in Figure 3.20. Unlike the previous linking, however, we now have more nodes to link between.

For these models we expect to see linking between the rooms by first just walls, in Model 3, then walls and doors in Model 4. If we look to the graph representation in Figure 3.22, we can see that all the nodes connect to each other via a wall edge, as they should. Room 1 can be seen to link to Room 2 and Room 4, as we would expect from the BIM. The Outer Zone can be seen connecting to each room by wall, window and door edges as appropriate.

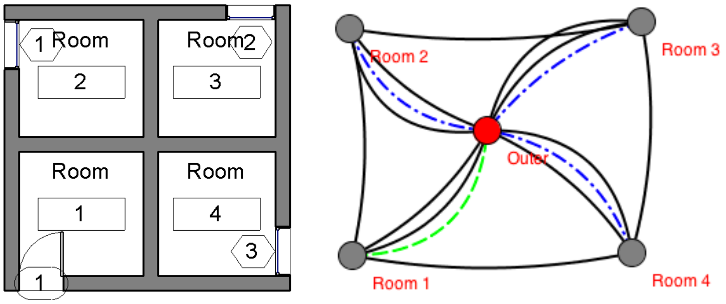


Figure 3.22: Model 3's floorplan beside its graph representation

In Figure 3.23 we can see the floorplan on the left with the additional internal doors. We would expect these doors to result in extra edges, each room connecting to two neighbours via a door edge, which the graph representation shows us is exactly what the system produces. The XML used to generate the graph representations for all these models can be checked in Appendix A.

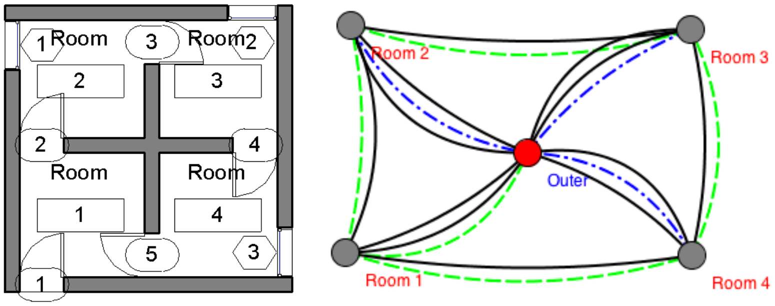


Figure 3.23: Model 4's floorplan beside its graph representation

In Model 5, the new element we're interested in is a zone which cannot be reached directly from the Outer zone. This will demonstrate that our system is not incorrectly linking zones to the Outer zone, and that the system is capable of creating more complex and realistic graphs requiring multiple steps to traverse. If we look at Figure 3.24 we can see the isolated zone, Room 9, in the middle of the graph linked by wall edges to its neighbours. We can see from a comparison of the floorplan and graph that it has all been linked correctly.

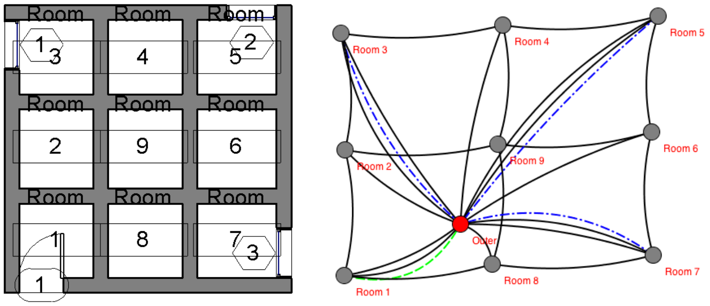


Figure 3.24: Model 5's floorplan beside its graph representation

From Model 5 to Model 6, the change is the addition of a series of internal doors, as shown earlier in Figure 3.17 on the right. If we look at the graph representation presented on the left of Figure 3.25, we can see this series of links depicted by the dashed green door edges. Again, the graph is as we would expect.

Model 7 is the same as Model 6 but removes the window edges, so we expect the same graph as we had for Model 6 but without the window edges. Comparing the two graphs in Figure 3.25 we see almost identical graphs, but the right hand graph representing Model 7 lacks the window edges. From the graph visualisations, both graphs are well formed and correct when compared to their BIMs.

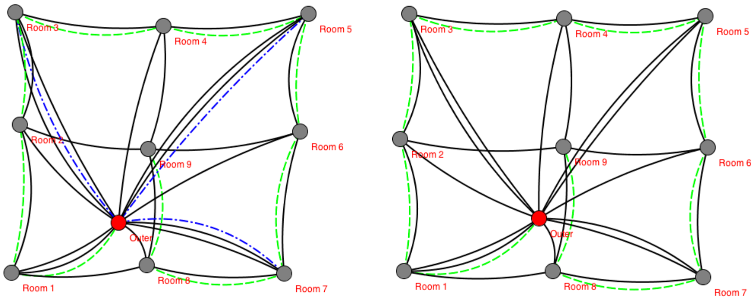


Figure 3.25: Model 6's graph representation, left, beside Model 7's graph representation, right

So far we've demonstrated our ability to generate wall, window and door edges along with room nodes. Using the multi-level "Simple Model", we can demonstrate the remaining edges: Conceptual, Floors, Stairs, Elevators and Elevation. We will examine this model below and relevant XML excerpts can be found in Figure 3.27 that help support the visual evidence we shall provide.

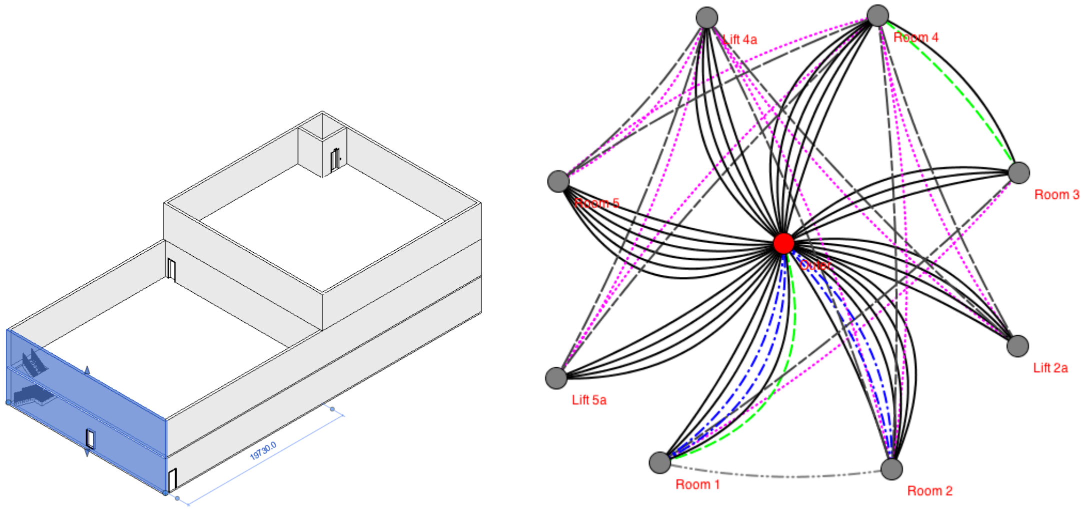


Figure 3.26: The Simple model above its graph representation

With this model we should see travel links between the rooms linked both by the stairs and also those linked by the elevator. These are indicated in Figure 3.26 by magenta dotted lines. The model itself is divided in half, with two rooms designated on each of the first two floors and a single room on the final floor that can only be accessed via the elevator.

Complete images of the floorplan and the full XML output for the graph can be found in Appendix B on page 140. The first floor of the building has no wall dividing Room 1 and Room 2, meaning the two will be linked by a conceptual separator edge. You can see this represented by a grey, dot-dot-dash line on the bottom of graph representation in Figure 3.26.

Looking to the graph representation, we should see a travel edge linking Room 1 and Room 3, represented by a magenta dotted line. This line represents the travel link created by the stairs and we can see it is present. We also expect to see Room 2, Room 4 and Room 5 connected by sequential travel edges, due to the elevator, but if we look at the graph we can see more than just the three expected travel edges.

The reason for this is, as discussed in 3.3.6, elevators are found via shaft matching. In this particular floorplan, the elevator shaft has been made into its own room. Because of the inaccuracies of bounding box matching, this results in a doubling up of linking, where the Elevator creates travel edges to both the room above as well as its shaft space, e.g. Room 2 will link to Room 4 and Lift 4a. This behaviour can be avoided simply by not labelling the Elevator shaft as a room.

simple6state.xml - Room 1 & Room 2

```xml
<zone0>
  <name>Room 1</name>
  <previous>Outer</previous>
  ...
  <conceptuals>
    <conceptual0>
      <neighbour>Room 2</neighbour>
    </conceptual0>
  </conceptuals>
  <travels>
    <travel0>
      <neighbour>Room 3</neighbour>
    </travel0>
  </travels>
  <roofs>
    <roof0>
      <neighbour>Room 3</neighbour>
    </roof0>
  </roofs>
  <floors>
  </floors>
  <environments>
  </environments>
</zone0>
```

```xml
<zone1>
  <name>Room 2</name>
  <previous>Outer</previous>
  ...
  <conceptuals>
    <conceptual0>
      <neighbour>Room 1</neighbour>
    </conceptual0>
  </conceptuals>
  <travels>
    <travel0>
      <neighbour>Room 4</neighbour>
    </travel0>
    <travel1>
      <neighbour>Lift 4a</neighbour>
    </travel1>
  </travels>
  <roofs>
    <roof0>
      <neighbour>Room 4</neighbour>
    </roof0>
    <roof1>
      <neighbour>Lift 4a</neighbour>
    </roof1>
  </roofs>
  <floors>
  </floors>
  <environments>
  </environments>
</zone1>
```

Figure 3.27: An edited excerpt of the XML representation of the multi-level graph, showing selected XML for the first two rooms

Within the graph representation and XML excerpt, we can also see the linking via the roof/floor edges. Room 3 lies above Room 1, so we link Room 1 to it via a Roof edge. If we look at the XML code for Room 1 shown on the left of Figure 3.27, we can see this edge represented.

This use of roof and floor edges allows us to capture vertical linking and to represent it cleanly in the XML, so it is easy to tell what room lies above and what room lies below. In the Graph representation these links are represented by a dark grey long dash-short dash line. We can see these links presented in Figure 3.26, with the rooms linked as expected.

Finally, we look to show the correct linking of the External Elevation edges between the Outer Zone and those zones above the first floor. If we look to the XML excerpt in Figure 3.28, we can see the entry for wall0, an external wall on the third floor. In the XML excerpts above we have generally redacted the materials to keep the tables small, but in this one we show them to demonstrate the External Elevation edges.

Considering walls on the first floor, if we look at their XML, we will find a single material entry: Brick. But for the external wall on Room 5, we can see that it has three entries; Brick, Height and Height. The two layers of height material capture the External Elevation edges, giving us a suitable method for measuring the cost of height.

simple6state.xml - Room 5

```
<zone4>
  <name>Room 5</name>
  <previous>Room 4</previous>
  <walls>
    <wall0>
      <material>brick</material>
      <material>height</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall0>
    ...
  </walls>
  ...
</zone4>
```

Figure 3.28: An edited excerpt of the XML representation of the multi-level Graph, showing selected XML for the first two rooms

3.4.2.1 Facility Scaling

Our system has also been tested on a more complicated facility based on the RAC_Advanced BIM that is distributed with Autodesk® Revit® 2012. To correctly map the facility we needed to add some internal zones, using the room tool, that were absent in some areas such as the second floor corridors. These missing corridor rooms would have caused our system to be unable to match rooms along the corridor and thus link those rooms to the outer zone.

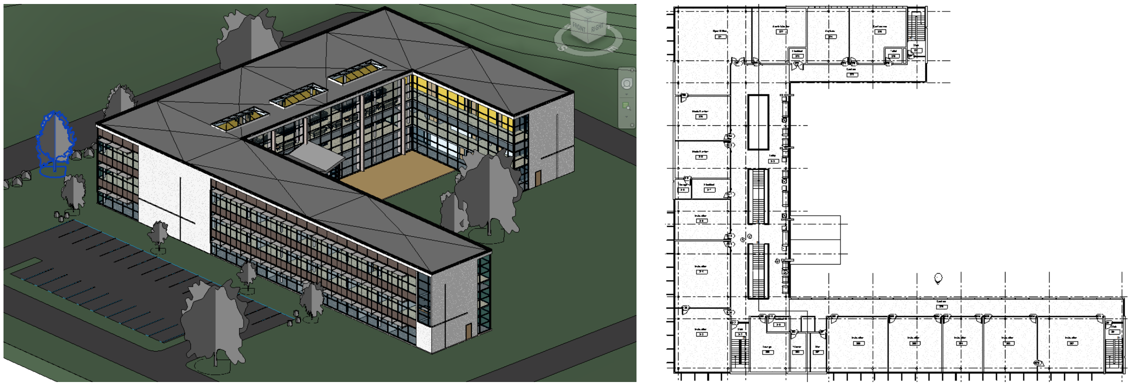


Figure 3.29: On the left, an external 3D representation of the RAC_Advnaced BIM. On the right, a copy of its third storey floor plan.

For a sense of the facilities complexity, part of its floorplan can be seen in Figure 3.29. Because of this complexity we will not attempt to show the XML, as it is both very long and very complex, but instead present a visual graph representation displayed in Figure 3.30. The Red Outer zone node can be seen at the center with the other nodes visible around it. The sheer number of edges makes their detail difficult to discern.

As you can see, our system is capable of scaling to a much more complex facility as long as all internal areas are mapped to rooms. The mappings are difficult to check completely due to the vast quantity of information, but we have performed spot checks to compare expected mapping to the generated and found no errors. While the processing time increases with scale, our system seems able to handle a facility of any size assuming sufficient processing time and memory.

We have demonstrated above that our software is capable of reading in a BIM and generating a graph from it. We have shown the ability to correctly generate a variety of edges under normal circumstances with appropriate linking to Zones. This demonstrates the efficacy of our software to abstract appropriate graph models from BIMs.

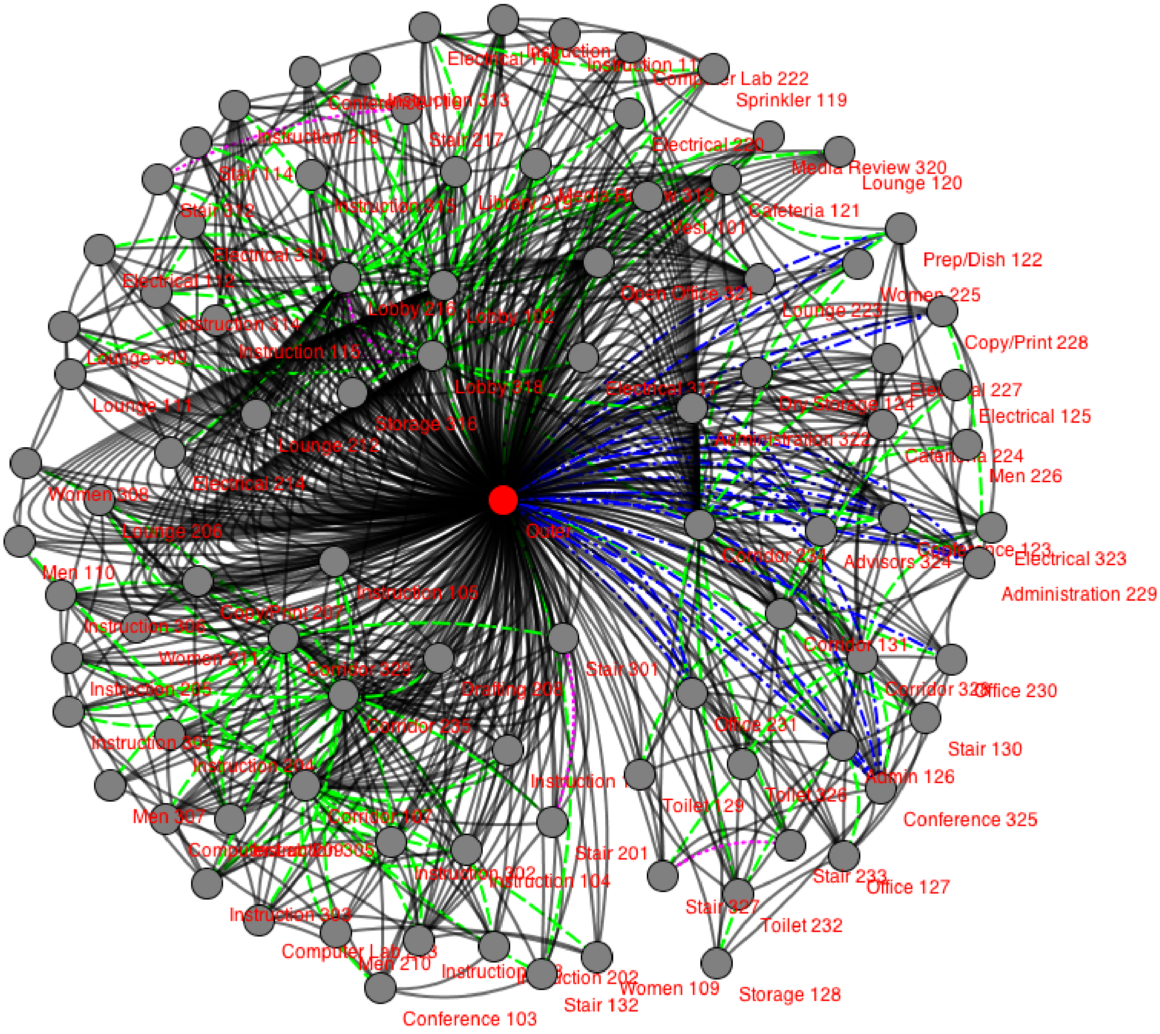


Figure 3.30: A graph representation of RAC_Advanced

3.5 Summary

In this chapter we have discussed a selection of toolsets available for BIM and the toolset most appropriate for our use. We have also discussed at length the implementation of our system for converting a BIM into a graph with useful nodes and edges. Finally, we presented results in support of our system, demonstrating its ability to map BIMs to suitable graphs. In the next chapter, we shall discuss the use of those graphs to implement security simulation.

Chapter 4

Security Simulation

Physical security is a complex problem to model and simulate. To begin with you need a large, accurate dataset to cover static elements such as barriers and sensors. Beyond this, you may also need to model dynamic data such as threat, communication, alarm assessment and response. You then need to map this data to an appropriate model to enable useful analysis.

Chapter 2 covered work that has gone into security simulation. In several of these simulations we have seen difficulties such as data input (Tarr, 1992, 1994; Tarr and Peaty, 1995) or limited path modelling (Garcia, 2001). As covered in Chapter 3, using data extracted from BIMs provides us with a method to mitigate both these problems.

In the following sections we will present details on our method for modelling elements of security. We will then present the results of our simulation along with a discussion of how to interpret them.

4.1 Method

Our methodology for modelling security is based upon the three Ds of security that we have discussed previously: Delay, Detect and Detain. We did consider Deterrence and believe it is an important aspect of security but decided not to address it in this preliminary work. We also take a cue from Tarr (1992, 1994); Tarr and Peaty (1995) in proposing that a model that is fundamentally secure is valid irregardless of how likely it is to deter attacks.

For a facility to be considered secure we would want to see it as being able to Delay an attacker from getting to and from their goal. We would want to see it have a high chance to Detect the attacker while they are trying to reach their goal. Finally, the two properties should work in tandem so that an attacker can be delayed long enough to be detected and an appropriate response implemented that results in their being Detained before they achieve their goal and escape.

4.1.1 Delay

In the previous chapter we covered the methods used to convert a BIM into a graph. Once it has been converted into our internal graph model, we are able to exploit graph theory to examine the delay characteristics of the BIM. To explore the vulnerabilities of a facility we assume a worse case scenario, our thinking being that if the facility can be hardened to improve its handling of a worse case attack then other attacks will necessarily be less dangerous.

To model this we use Djikstra's algorithm (Dijkstra, 1959), a well known method for finding the shortest paths within graphs. It uses weighted edges to find the cheapest path between nodes within a graph from a starting node. As discussed in the previous chapter, we include in our modelling an Outer Zone to act as an initial point of ingress for any simulated attack and it serves as our initial node for use with Djikstra's algorithm.

For our Delay modelling we take the generated graph and use Dijkstra's algorithm to calculate the shortest or cheapest path from the Outer Zone to all other areas within the facility. The weight we apply to each edge is the cost for an attacker to travel along it in terms of Delay. In this way we can quickly calculate the worst case scenario for an attacker trying to breach any zone within a modelled facility.

The difficulty in achieving this modelling was converting the elements of a facility such as walls and doors into appropriate delay values. To achieve this we needed to extract information from the BIM on what an element was composed of and then match that to an appropriate delay value. This is further complicated when you consider that different materials will offer different delay values against different attacks.

To deal with this we first designed a set of objects to model what we considered the relevant attributes of delay. We then developed a system to capture the attributes from BIM by matching materials to a defined dataset. We then expanded this system to better simulate the delay an attacker is likely to encounter.

4.1.1.1 Modelling Delay

As explained in Chapter 3 we convert a BIM into a graph. The graph is composed of nodes, representing rooms and areas, and edges, representing possible paths. By weighting those edges with their delay values we can then establish the quickest path an attacker might take from one location to another.

To represent this in software we designed a set of objects to model a facility and capture the behaviour. This allows us to map parts of the facility to objects and allows for future work at a high fidelity. As an example, if you think of a window, wall and a door, they are all barriers and would be mapped to an edge.

For a simple shortest path calculation, we do not need to differentiate between them, they could be captured by an integer. By mapping them to objects each barrier can contain its own properties and methods specific to it, enabling nuanced future simulations. For instance, virtual red teaming.

In virtual red teaming you would have agents attempting to breach a facility, called the red team, while other agents attempt to defend the facility, the blue team. By differentiating between door and window objects now, we make it easier to add behaviour such as looking through a window or picking the lock on a door later. Because a window would be its own custom object we could add properties and methods to allow looking through it, while a wall would lack that functionality.

Even at our present level of simulation, this added complexity has advantages. By modelling more objects which are named for their real world counterparts, it can be easier to understand and track an attacks progress through a facility. The objects also contain details necessary for creating the attack's shortest path integers, such as material.

The material contains a list of attacks that are effective against it along with their relevant cost. This means the material can be queried about a given attack and respond with that attacks cost. This value can then be assigned to the barrier, providing a cost for that edge to be traversed by the shortest path algorithm. The objects also allow us to compartmentalise functionality and create a logical hierarchy.

By our naming convention, classes are named as ObjectnameClass while an instantiated object may be referred to simply by Objectname. Below we may refer to a ZoneClass or Zone object, depending on context, but essentially they are both the same and may be considered equivalent.

An overview of our objects can be seen in Figure 4.1. The ZoneHandler object can contain many zones and each zone can contain many barriers. Each barrier can link to two zones and can contain many materials[1].

---

[1] This functionality was added later, originally they could contain only one material

The ZoneHandler object acts as a controller, containing all the other objects and performing the graph analysis. The Zone objects represent areas such as rooms and form the nodes of our graphs. The Barrier objects represent the various possible links between areas and provide the edges for our graph analysis.

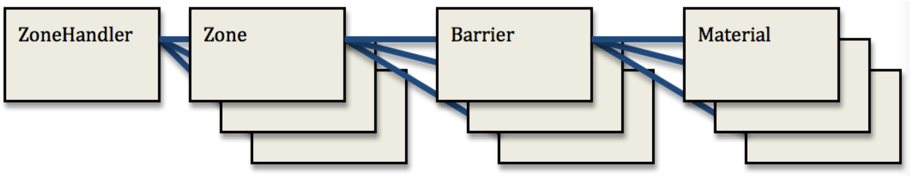


Figure 4.1: A view of our object hierarchy

The BarrierClass is a abstract class and is itself never instantiated, but provides the basic attributes and functions necessary to model the edges. We have a large collection of child objects inheriting behaviour, as can be seen in Figure 4.2, each designed to capture slightly different behaviour. Many of the objects are functionally very similar at this time but allow us to more accurately represent a building and allow for future subtleties to be captured, such as windows being seen through thus allowing an attacker to be spotted.

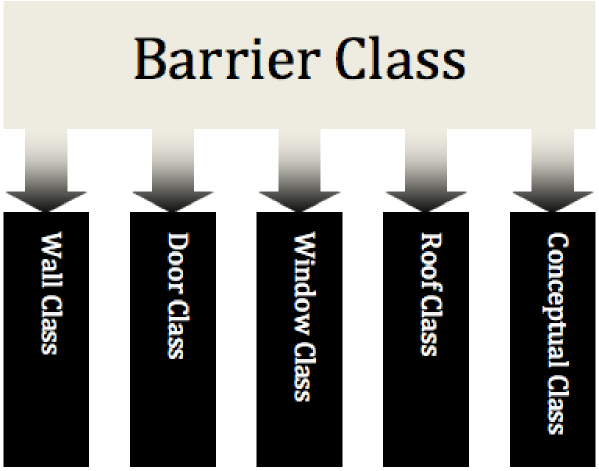
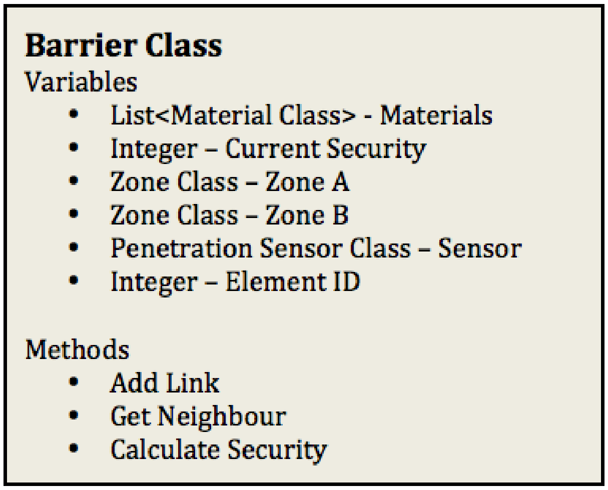


Figure 4.2: The Barrier Class object and its children

The Barrier Class itself can be seen in Figure 4.3 and we will here discuss its attributes. The first item is a Material Class object, which contains a string giving the name of the material and a list of Attack Class objects. The Attack Class objects provide a list of tools required for the attack and average time for that attack to succeed, capturing the interaction of tools and materials in an object.

So if the material was for instance “wood”, we could list all available attacks that can be used against wood such as “breaking with rock”, “cutting with axe” and so on. For "cutting with axe" the Attack Class would then contain a list of tools, in this case "Axe", and the average time for the attack to succeed along with its standard deviation. This allows us to weight the barrier with the appropriate delay when we call the Calculate Security method, which is discussed below.



Barrier Class

Variables

- List<Material Class> - Materials
- Integer – Current Security
- Zone Class – Zone A
- Zone Class – Zone B
- Penetration Sensor Class – Sensor
- Integer – Element ID

Methods

- Add Link
- Get Neighbour
- Calculate Security

Figure 4.3: The Barrier Class object and its important variables and methods

The current security integer begins simulation set to an invalid value, -1, and is updated after Calculate Security method is called with the effective delay value of the current barrier instance. Zone A and Zone B allow the Barrier to link to the zones present on either side of it. These zone references are useful during the graph traversal and they are utilised by the Get Neighbour method call, which we will detail below.

Finally we have a Penetration Sensor Class object which will be covered in more detail in Section 4.1.2 and a Element ID integer. Within Revit®, all objects in a BIM have a reference ID or Element ID. We use the Element ID field on the Barrier Class to allow us to capture this value so we can reference back to the object within the BIM, allowing for easier isolation of problem barriers by searching the BIM for their Element ID.

The Add Link method accepts one or two Zone Class objects as part of its call. When it's provided with a Zone Class object, it checks to make sure that the Zone Class isn't already assigned to this Barrier to avoid complications caused by objects such as partial walls within a zone that could lead to an infinite loop during graph traversal. If the Zone Class is not already assigned to Zone A or Zone B, then Add Link sets the provided Zone Class to whichever variable is available.

As an example, if we have a Barrier object named "Wall" and wish to associate it with two Zone objects, "Lobby" and "Office" we could call Wall.AddLink(Lobby, Office). The code for AddLink will then attempt to assign Lobby to Wall.ZoneA which, finding it empty, will succeed. It will then attempt to assign Office but on attempting to do so, will find a zone already assigned and compare the Office and Lobby objects to make sure they are not the same.

Once it has been established that the two objects represent different Zones, it will then attempt to assign Office to Wall.ZoneA. As ZoneA already contains a value, Lobby, this will fail and it will attempt to assign Office to Wall.ZoneB instead, which will succeed. If AddLink could not assign both Zones it was passed, perhaps Wall.ZoneB was already assigned to "Courtyard", then the method would return an error.

The Get Neighbour method takes a Zone Class object as an argument. It checks Zone A and Zone B to see which, if either, matches the provided Zone Class. If one matches, the method will return the other, allowing a Zone to call this method on its barriers to identify its neighbours so it can find which has the least delay and update as necessary. If neither Zone A or Zone B is a match, the method returns a null to indicate an error.

So if we used the Wall object above and called Wall.GetNeighbour(Office), the method would return the Lobby object. If we instead called the method with Wall.GetNeighbour(Courtyard), it would return a null as neither Zone A or Zone B is a match for "Courtyard". This allows us to quickly establish adjacency.

Calculate Security is the main method for the Barrier Class and its children, with pseudocode provided in Figure 4.4. It accepts a list of Tool Class objects, which describe items that might be used as part of an attack. The Calculate Security method then calls a method on the Material Class object of the barrier, called Attack With.

```
CalculateSecurity(ToolList)
  CREATE AvailableAttacks
  CREATE BestAttack
  FOR EACH Material IN Barrier
    AvailableAttacks = Material.attackWith(ToolList)
    Material.CalculateSecurity(AvailableAttacks)
    BestAttack = Material.AttackUsed
    IF BestAttack IS NULL
      Barrier.Security = VeryHighValue
    ELSE
      Security = BestAttack.mean
```

Figure 4.4: Pseudocode for the Calculate Security method

The Attack With method takes in a list of tools and searches the list of Attack Class objects on the material to find those which can be accomplished with the provided list of tools. It collects all possible attacks and returns them to the calling method, which then calls the Calculate Security method on the Material Class object with the provided list of attacks. If there is at least one attack in the provided list, the attack with the lowest delay will be remembered by the Material Class object and its delay value can be set as the delay of the barrier.

Should no valid attacks be found against the material from the list of available tools, then the material sets an arbitrarily high delay value. This arbitrarily high value can then be set as the delay of the barrier, reducing the likelihood that this barrier will be used as a link between zones. In any case where two attacks result in the same delay on a material, the latter attack will be set as the one used against the barrier.

This rule holds true throughout the system we developed, meaning that should two barriers have the same delay value, when finding a path the system will only indicate use of the latter. This does introduce the slight risk of a material being hardened against one attack when a different attack could still bypass it or one door being hardened when another will prove just as weak. This risk is mitigated by the fact the system is intended to be iterative and any changes made can be simulated against to uncover remaining weaknesses.

The use of computational simulation allows us to cheaply repeat our tests with all the possible combination of attacks. The intention of this behaviour is to help isolate specific attack combinations which pose a high risk so they may be hardened against. The system iterates through the possible combinations of tools from those provided to exhaustively test the problem space.

The system records the state of the graph after each simulation along with the attacks used to create that state. At the end of a complete run, our system performs analysis across all recorded states to establish the mean breach time required and standard deviations. This information is presented in the security report at the end of the run.

4.1.1.2 Matching and Delay Value Import

The system described above is reliant upon having delay values to use. Using appropriate delay values was a more complex problem than we anticipated, as we needed realistic values to use and a way to ensure we were using the correct values in the correct situation. The first problem we address with the use of the delay values provided by Alach (2007).

Attacks	Materials
Rock	Glass
Hammer	Earth
Axe	Wood
Oxy Cutting	Brick
Electric Drill	Concrete
Explosive	Steel

Figure 4.5: The six basic attacks and materials we took from Alach

In Figure 4.5 we provide a list of the basic attacks and materials taken from the work by Alach (2007). In Figure 2.10 on page 29 you can see a more complete list, showing each attack paired against all six materials and their mean penetration times as established by Alach (2007). Using this data from Alach gives us a set of delay values we can apply to our barriers.

To apply the appropriate delay values in our simulation we must select the appropriate attacks and the appropriate material. The attacks are set at the time of the run but at present may only be altered by changing the relevant call in the code itself. While a simple User Interface (UI) might be introduced in future, unfortunately for now the attacks can only be changed by recompiling the extension.

Materials are assigned to the barriers during the object generation process, previously described in Chapter 3. As the barrier is created the BIM element is interrogated to retrieve a list of the materials it is made from. A match method then moves through each material on the element and attempts to find a match from those materials created from the data provided by Alach (2007).

The matching method is simple, using a string comparison to try and match materials from the BIM with the six created from Alach (2007)'s values. It takes the name of the material provided by the BIM and checks to see if the name of any defined material is contained within it, ignoring case and extraneous characters. Should it find such a match, it assigns the defined material values to the Barrier object.

As an example, the "Double Brick - 270" wall defined within Revit® contains the following layers within its structure:

- Masonry - Brick - Brown
- Air Barrier - Air Infiltration Barrier

- Layers Above Wrap
- Masonry
- Layers Below Wrap

When matching, the method will compare each of these to Glass, Earth, Wood, Brick, Concrete and Steel. The first layer, “Masonry - Brick - Brown”, will be matched against a defined material object as it contains the word “Brick”. No other layer contains a word that can be matched, so the values for the brick material will be assigned to the Wall Object the system is creating.

Because of the currently limited scope of materials we have defined versus the materials that might be present in a real BIM, we also have default materials to assign to barriers. Walls are by default assigned brick, windows glass and doors wood. This allows us to simulate security against even incomplete facility designs, but all instances of a default material being assigned are flagged in a log that is output at runtime.

The log has been useful during development to locate instances where objects that should be matching were not due to bugs in the code. We foresee it continuing to be useful in this role should other materials be defined, as it will help isolate minor errors which are preventing a match. The log also outputs information on successful assignments of materials and links established, so can be useful for tracing the conversion process, though its present form requires some effort to parse by hand.

About midway through our development we considered the limitation of the delay values we had access to and the versatility users might want. For instance, a security company might have proprietary information on how long it takes to breach a particular brand of door that they would wish to model so they can show how it alters the security of a facility. To allow for this we needed to develop a method for importing user definable delay values.

To achieve this we decided to use eXtensible Markup Language (XML), a human and machine readable data format. We designed and implemented a file structure that would allow a user to implement new materials with reference to any valid attacks. We then implemented all of our standard materials and attacks within the file, giving users a central file to alter, granting them full control over available attacks and relevant delay values in response to those attacks.

The files form can be seen laid out in Figure 4.6. XML requires elements be delimited by a matching start and end tag, with nesting of elements supported. As can be seen, we cap the entire data set with a Cerberus tag, a reference to our internal name for the system which is a homage to the mythical guard dog of the underworld in Greek mythology.

materials.xml - General Form

```
<Cerberus>
    <Attacks>
        <Attack>
            <Name>Attack Name </Name >
        </Attack>
        ...
    </Attacks>
    <Materials>
        <Material>
            <Name>Material Name </Name>
            <Attacks>
                <Attack>
                    <Name>Attack Name</Name>
                    <Delay>Delay Value</Delay>
                </Attack>
                ...
            </Attacks>
        </Material>
        ...
    </Materials>
</Cerberus>
```

Figure 4.6: The layout of the xml file used for material import, with place holder names used in this example. A copy of our actual materials.xml file can be found in Appendix C

The file is read in at runtime before the BIM is converted into a graph. At the beginning of the file a list of possible attacks must be defined. This is so the relevant tool class and attack class objects can be generated in readiness to be attributed to the relevant material class objects.

After this the file defines the materials. Each material is defined by its name and then a list of relevant attacks providing attack name and delay values. A full copy of materials.xml as used during our delay simulations can be seen in Appendix C.

4.1.1.3 Height and Layers

Late in the development process we considered the problem of height, such as might be encountered by an attacker looking to breach a second storey window. Height introduces an extra layer of delay which presented several interesting problems. Firstly, elements located off the ground can only be reached by a select number of methods and secondly, delay arising from height would need to be added to existing delay values.

The first problem we overcame with our existing material/attack system. We created a new material, “Height”, then defined the attacks available for overcoming it. We introduced a “ladder” and a “climb” attack which are only used against height.

The second problem was more complex because it raised the issue of how we should model this particular situation. Our existing attack/delay system only allowed for one material per barrier and we had no existing model that would allow for the introduction of a height barrier between a second storey window and an attacker. We came up with two possible solutions, the “outer stack” method and the “window glaze” method. A diagram of the two can be seen in Figure 4.7.

Under the outer stack solution we would detect the number of levels within a BIM and generate extra zones for each level. Each of these zones would be linked back to the outer zone so that they form a stack. Any external barrier can then be linked to the outer zone at the same level.

The major advantage of this method is it requires minimal changes to how the system functioned. It is also logical as once an attacker reaches a certain outer level, it can attempt to breach any external facing zone on that level. This could also lead to easy to read attack logs: “Attacker reached level 2 with ladder. Attacker reached Room 201 via window with stone.”

Under the window glaze approach, rather than introduce extra zones we add a height delay to each relevant barrier. This introduces a challenge as we either need to calculate the height delay and apply it when we detect we’re working with any barrier about ground level or model it when we model the barrier at that level. To model it with the barrier required modifying our existing attack/delay system to support multiple materials for each barrier.

Inspite of the added complexity of its implementation, it was decided that the window

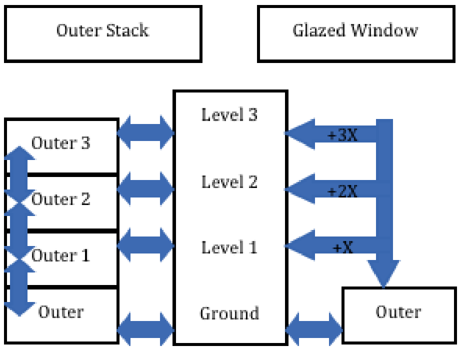


Figure 4.7: A diagram of Outer Stack vs Glazed Window

glaze method would be beneficial to the system. While the multiple materials aspect would require several changes to the existing system, it would improve the flexibility in the future. It would also result in more representative simulations.

Instead of worrying that the last material matched might not be the best representation of a barrier, now all materials within the barrier can be matched individually. Some materials, such as reinforced concrete, can also be more accurately modelled as multiple layers. While a hammer can be effective against concrete, it works poorly for the rebar reinforcing it and the inverse is true for a tool such as an oxy cutter.

By allowing for a wall to be modelled as its constituent parts, the system can select the best tool to use against each material. So with the reinforced concrete, the system can use a hammer to break the concrete and the oxy cutter to slice through the bars. To achieve this we implemented a material list within the barrier object and when performing a security simulation we run the provided tools against each material in turn and sum the resultant delay.

4.1.2 Detect & Detain

Detect and Detain are two other important aspects of vulnerability assessment. Detection uses a simple method described below, but Detainment proved more problematic. In the end we did not model Detain and will discuss the reason for this below.

To model Detain you need to integrate the response force times. While in the simple case this could be done with an arbitrary response time, we considered that a user could simply compare a known response time to the returned delay times to see if they were suitable. To model Detain in the more complex manner, considering relative locations of possible responders, alarm assessment and force capability requires the introduction of various factors considered too complex to include within the scope of this work.

In Chapter 5 under future work we will discuss the possibility of extending this work with the introduction of intelligent agents. Intelligent agents offer many interesting possibilities for extending this work, some of which has already been explored by Porter *et al.* (2014). For now we will address our work on Detect.

4.1.2.1 Modelling Detection

We chose to model detection such that any sensor that is passed is considered triggered. We ignored both false positives and false negatives, where an alarm might trigger an unnecessary response or might not trigger in response to an attack i.e. assume perfect sensors. Our reason for this was that it would otherwise need probabilities for each parameter, for each type of sensor that are available, vastly increasing complexity.

While most sensors can be rated on likelihood of detection or number of false positives over a time period, we felt to do this accurately would not be entirely compatible with our modelling of delay. Our system is built around the concept of quantitative analysis, by way of simulating numerous runs and averaging values from across those runs for analysis. Adding in an unreliable detection element would produce inconsistent results.

As an example, a motion sensor could be assumed to have a 58%(Osman *et al.*, 2009) chance of detection, given reasonable environmental conditions. If our list of attacks causes us to run 64 times, against which of those runs would we assign the failed detections? We decided it was better to assume 100% accuracy for sensors and simply provide an indication of the variance in time between the first sensor being triggered and a relevant zone being reached.

This increases the burden on the designer and future work might address this. We suggest for single path runs that the system could provide a list of detection points and indicate the combined likelihood of detection along the path based on provided data for the sensors. At present we model two classes of sensors, penetration and volumetric, as can be seen in Figure 4.8. Other classes of sensors exist, such as line and point, but for proving our concept system we selected these two sensor models to begin with.

Penetration Sensors are items such as reed sensors, attached to doors or windows and triggered by the relevant object being disturbed. Volumetric Sensors include items such

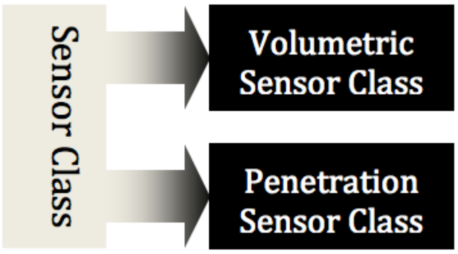


Figure 4.8: The abstract Sensor Class and its two children

as video cameras and motion sensors. In terms of functionality they are both treated quite similarly at this time, so most of the functionality is defined within the parent Sensor Class.

The next step to modelling detection was locating the sensors within the graph so that they can be considered. In the end, we found this required two different approaches, one for volumetric and one for penetration. This was because of the difficulty of modelling the items within BIMs.

When we looked into modelling penetration sensors, we could not find established models and methods for doing so. While objects exist that can be used for adding volumetric sensors to a model there is no equivalent for its counterpart. We therefore created our own solution.

We looked into the information available within BIM that can be easily altered. Within Revit®, objects have Identity data including a comments field that can be easily written to during BIM design. To add penetration sensors we implemented a system where you simply need to add the keyword “sensor” to the comment field, allowing for information such as “SecuriCompany reed sensor MK447gh” to be usefully interpreted by a user and our system.

The implementation for this requires that we go through all door and window elements at the end of our graph generation. The pseudo code for doors can be seen in Figure 4.9. For each door and window we check the comment parameter for the sensor key word. If the key word is found, we then create a penetration sensor object and use the element ID of the door or window to attach it to the correct object.

```
COLLECT ALL DoorElements
FOR EACH DoorElement
  FIND DoorObject USING DoorElement.ElementID
  FOR EACH Parameter IN DoorElement
    IF Parameter.Definition.Name IS "Comments"
      IF Parameter.AsString CONTAINS "SENSOR"
        CREATE PenetrationSensorObject
        DoorObject ADD PenetrationSensorObject
```

Figure 4.9: Psuedocode to find penetration sensors and assign them to Doors

Volumetric sensors are items such as cameras and motion sensors. As can be seen in Figure 4.10 they derive their name from the volume of area they cover. While some such sensors can rotate or pan, most are set to cover a fixed area.

While we have included cameras as a volumetric sensor, from a security stand point they are quite dissimilar to devices such as motion detectors in that their detection chance is much less certain and dependant upon human operators. Once detection probability is modeled, we propose cameras could have their detection chance modeled as DP/(OD/CPO)[2]. Further research into appropriate metrics would be warranted at that point but for now for the purposes of our simulation we will treat them as a volumetric sensor.

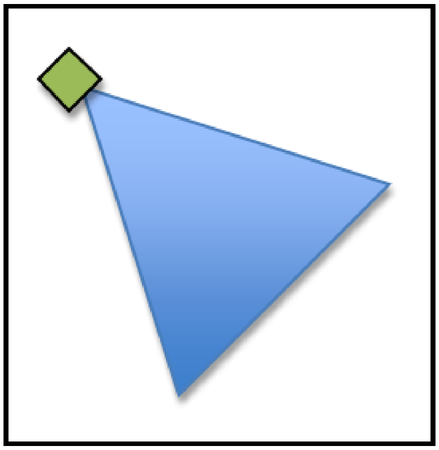


Figure 4.10: An example volumetric sensor in a room. The green box indicates the location of the camera or motion sensor, the blue area indicates the volume of area covered by it.

---

[2] DP being Detection Probability of the camera, OD being probability of the Operator Detecting an incident and CPO being the number of Cameras Per Operator

To acquire the volumetric sensors, we create a list of objects defined as Security devices. Within that list, we search for elements whose name or family name contains the keyword "camera" to give us a list of elements we can be confident qualify as volumetric sensors. From that element we create a new Volumetric Sensor Class object and capture the central 3D or xyz point of the element to use in placing it.

Once we have the central point, we captured a list of all rooms and check through them to isolate those that might contain it. Once we have identified a room whose bounding box contains the central point, we identify the Zone Class object made from it and assign the volumetric sensor to it. The pseudo code for this can be seen in Figure 4.11.

```
COLLECT ALL SecurityElements
FOR EACH SecurityElement
  IF SecurityElement.Name OR SecurityElement.Family.Name CONTAINS "camera"
    CREATE VolumetricSensor
    CREATE camPoint FROM SecurityElement.BoundingBox
    COLLECT ALL RoomElements
    FOR EACH RoomElements
      CREATE RoomBound FROM RoomElement.BoundingBox
      IF RoomBound.Z.min <= camPoint AND RoomBound.Z.max >= camPoint
        IF RoomBound.X.min <= camPoint AND RoomBound.X.max >= camPoint
          IF RoomBound.Y.min <= camPoint AND RoomBound.Y.max >= camPoint
            FIND ZoneObject USING RoomElement.ElementID
            ZoneObject ADD VolumetricSensor
```

Figure 4.11: Psuedocode for finding volumetric sensors and assigning them

During simulation we make use of the sensor objects to establish a Time Since Detection (TSD) which can be analysed across runs to provide feedback to a user. As the graph is traversed, when a barrier is selected as the easiest path, it is checked for a penetration sensor. If a penetration sensor is found, then the Zone that is about to be entered is assigned a TSD equal to the cost of the barrier, the assumption being that the sensor would be triggered at the start of the breach attempt. This most closely resembles the behavior of a glass break sensor, triggering at the point of break, and in future we would look to include the option of triggering before or after breach.

If the Zone that is being entered has a volumetric sensor, then the TSD is set to 0 on the zone. Any room accessed from that room will check if the preceeding room has a TSD and, if it has a value of 0 or greater, it will add the barrier cost to the previous room's TSD and assign it to itself. In this manner once the graph has been fully traversed and the shortest path to each zone found, we can also see which of those paths result in alarms and how long to reach each zone once the alarm has occurred.

4.2 Results

In the previous section we described our methods for modelling Delay and Detect. In this section we will present the results of our method to demonstrate the functionality of our system, generated from a selection of models.

4.2.1 Delay - Model 1

Figure 4.12: Model1 through Model6 as detailed in Chapter 3

We would expect when assessing security that it is quickest for an attacker to breach a window before a door and a door before a wall. This comes down primarily to the materials they are typically made from, as indicated by the values provided by Alach (2007). In terms of time, it is cheaper to break down a wooden door than a brick wall and a glass window is cheaper again.

To test our security simulation we provided our software with the list of materials and attacks provided in Appendix C. When we run the system, we use a slightly restricted list of tools; Hammer, Axe, Electric Drill, Rock, Climb and Ladder. We chose these tools to provide a good range of values and because tools such as Explosives make breaching any material trivial.

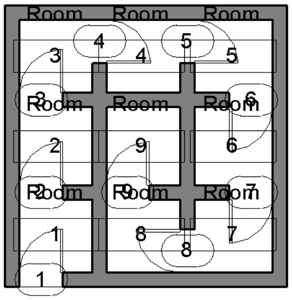
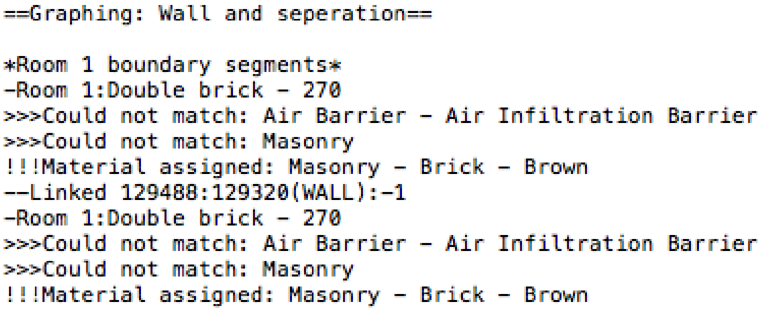


Figure 4.13: Model7 as detailed in Chapter 3

The results of the conversion process for turning the Model1 BIM into a graph can be seen in Appendix D. A small excerpt is shown in Figure 4.14, where we can see the system work its way through the wall boundary of the room. We can see that when examining the wall the system finds three layers of material but can only match "Masonry - Brick - Brown", providing feedback to the user on materials that cannot currently be accounted for so they may make decisions about defining or altering them.



```
==Graphing: Wall and seperation==

*Room 1 boundary segments*
-Room 1:Double brick - 270
>>>Could not match: Air Barrier - Air Infiltration Barrier
>>>Could not match: Masonry
!!!Material assigned: Masonry - Brick - Brown
--Linked 129488:129320(WALL):-1
-Room 1:Double brick - 270
>>>Could not match: Air Barrier - Air Infiltration Barrier
>>>Could not match: Masonry
!!!Material assigned: Masonry - Brick - Brown
```

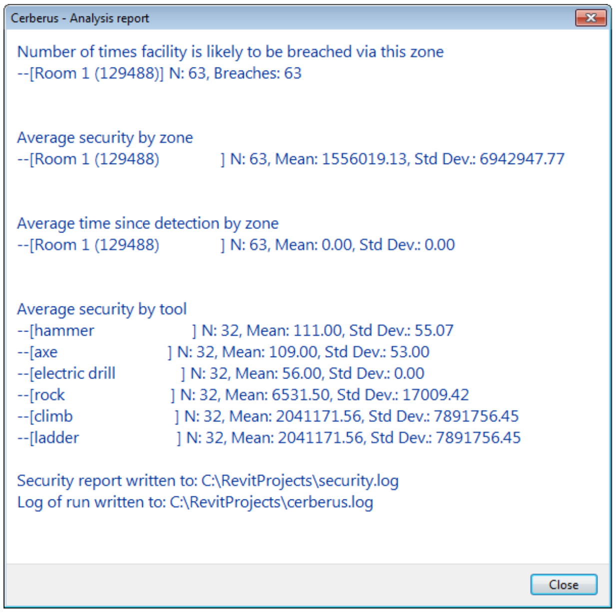
Figure 4.14: Excerpt of the log produced by a run on Model1

When the system runs it takes a list of provided tools and runs all permutations of those tools agains the provided model. In this way we can explore the entire threat space and begin providing feedback to the user. So, given a hammer, rock and ladder it would run the following permutations:

- Hammer
- Hammer Rock
- Hammer Rock Ladder
- Hammer Ladder
- Rock
- Rock Ladder
- Ladder

When the system runs each zone is examined and assigned a security value, essentially the delay value or length of time required to reach that zone. Each zone also records the zone it was reached from, providing a chain of zones that indicate the best way to reach any given zone. Finally the zones also record their breach point, allowing the system to remember whether a zone was reached via a door, window, wall or duct.

The system saves a copy of the graph after each of these runs, capturing what we refer to as its state. Once all permutations have been run, the system will go back through the available states and perform statistical analysis so that information can be provided on each tool and each room. For Model1 we will get the results shown in Figure 4.15.



```
Number of times facility is likely to be breached via this zone
--[Room 1 (129488)] N: 63, Breaches: 63


Average security by zone
--[Room 1 (129488)          ] N: 63, Mean: 1556019.13, Std Dev.: 6942947.77


Average time since detection by zone
--[Room 1 (129488)          ] N: 63, Mean: 0.00, Std Dev.: 0.00


Average security by tool
--[hammer                ] N: 32, Mean: 111.00, Std Dev.: 55.07
--[axe                ] N: 32, Mean: 109.00, Std Dev.: 53.00
--[electric drill       ] N: 32, Mean: 56.00, Std Dev.: 0.00
--[rock               ] N: 32, Mean: 6531.50, Std Dev.: 17009.42
--[climb              ] N: 32, Mean: 2041171.56, Std Dev.: 7891756.45
--[ladder             ] N: 32, Mean: 2041171.56, Std Dev.: 7891756.45

Security report written to: C:\RevitProjects\security.log
Log of run written to: C:\RevitProjects\cerberus.log
```

Figure 4.15: Results from the system running against Model1, delay values in seconds

Our N value represents the number of simulations involved. So if an entry has "N: 63", then there were 63 total simulations compared to reach that result. N will typically represent the possible combinations of tools, so with 3 tools we would expect an N value of 7 as there are 7 possible permutations.

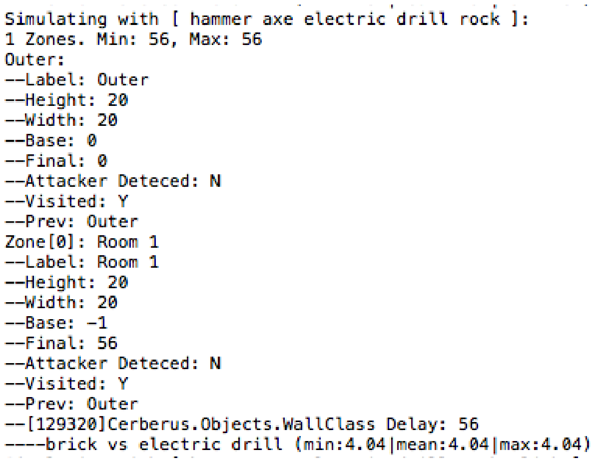
Looking at Figure 4.15, most of the average values involve an N of 63, due to the 63 combinations possible from the 6 tools. The average security by tool N values are nearly half of this at 32. This is because when looking at the tool's impact, we only examine the subset of simulations where that tool was one of the ones used.

The number of times the facility is likely to be breached via this zone indicates, out of all the runs with all permutations of tools, how often the specified zones previous zone is the Outer Zone. This is intended to provide a measure of vulnerability, with rooms with a high count being accessed regularly from outside. This could indicate that the room is a weak point and should be hardened or have sensors added.

The number of breaches can be compared to the mean security of the zone (Average security by zone). If the mean security is high as are the breaches then it is most likely that while the zone is often accessed from outside, this is only because there is no cheaper path to it. Looking at our results for Model1 we can see that Room 1 is always breached (N: 63), as we would expect given it is the only room.

Its mean security is high (1556019.13) but so is the standard deviation(6942947.77), suggesting that the room is quite secure, but more vulnerable to certain attacks than others. We can examine the impact of various tools by looking at the “Average Security by tool” list. We can see Climbing and Ladders are both particularly ineffective and are most likely distorting our values, so we might want to run again without them. At the same time, Hammer and Electric Drill are both reasonably effective so we may wish to consider our materials and look to harden against those attacks.

All of this information together is intended to give an operator a feel for the facility they are simulating. We can also examine the security log for this model, generated at runtime, which will highlight what materials were matched and what attacks were used. The security log is extensive and we will include it in its entirety in Appendix E. Below is a small excerpt.



```
Simulating with [ hammer axe electric drill rock ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
```

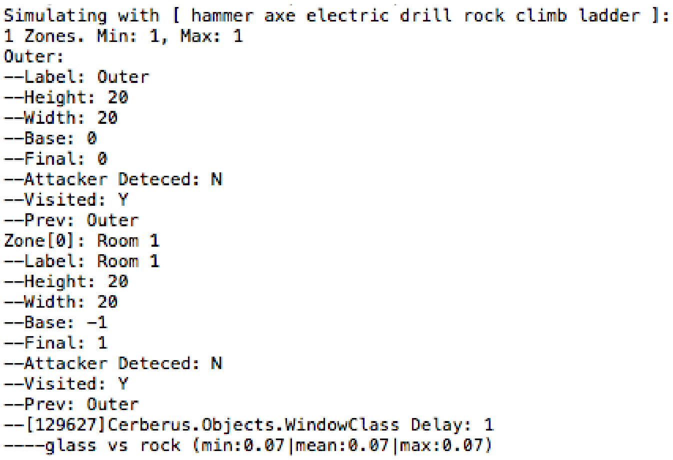
Figure 4.16: An excerpt of Model1’s security log

We can see at the bottom of Figure 4.16 that the previous zone to Room 1 is the Outer zone (Prev: Outer). Below this, we see that this zone was reached via a Wall Class object which matches Element ID 129320 at a delay cost of 56 seconds. We can see that this was achieved by using the Electric Drill against the brick, which has a mean breach time of 4.04 according to Alach (2007), before adjustment.

We can see from the above that against an admittedly very simple model our system works and produces results we have demonstrated to be informative. We will look at the other models now, in the context of what aspect of the system they were designed to test or demonstrate and show how the system performed against these challenges. Beginning with Model2, which was designed to show the system will select the cheapest path available among many.

4.2.2 Delay - Model 2

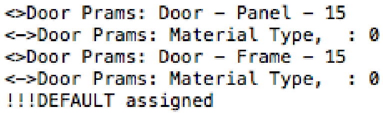
In Model2 a door and three windows were introduced. Given almost any collection of tools we would expect the windows to be the easiest breach point, with the system referencing the last window it examines. If we look at an excerpt of its security log in Figure 4.17, we can see this plays out as we would expect.



```
Simulating with [ hammer axe electric drill rock climb ladder ]:
1 Zones. Min: 1, Max: 1
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 1
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129627]Cerberus.Objects.WindowClass Delay: 1
----glass vs rock (min:0.07|mean:0.07|max:0.07)
```

Figure 4.17: An excerpt of the Model2 security log

The zone is breached via the Window Class object as it provides the cheapest path. With several of the tools providing similar breach times, it retains the last one during processing, which in this case is Rock. If we look at the BIM to graph conversion log for Model2 we find that the wood in the door is a default assignment, as previously described, with the relevant excerpt in Figure 4.18.



```
<>Door Prams: Door - Panel - 15
<->Door Prams: Material Type,  : 0
<>Door Prams: Door - Frame - 15
<->Door Prams: Material Type,  : 0
!!!DEFAULT assigned
```

Figure 4.18: An example of default material assignment from Model2's conversion log.

4.2.3 Delay - Model 3 and Model 4

Model3 and Model4 are both intended to show that the system works with multi-room designs. Model3 was intended to show that the system will access rooms through the easier external links rather than the more costly room to room links via walls. However, much the same can be seen from Model4 as we will show so we shall exclude Model3 from the discussion here.

In Model4 we provide internal doors as well as external doors and windows. When this model was designed it was thought that the system would present an interesting path through the facility, providing a breach in one room then using it to gain access to others. However, that is not the case.

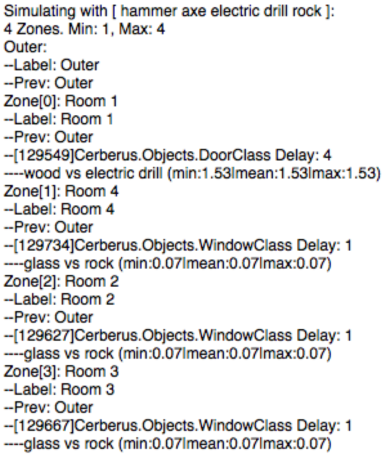
Counter to our own intuition, which is no doubt founded in the habit of finding a single entrance then navigating an internal space, the path from the outside is always cheaper. To access Room 3 from the outside you need to only breach a single window. To access Room 3 from Room 2 or Room 4 you must breach a window and a door, the delay times compounding.

As can be seen from the security excerpt in Figure 4.19, each room is reached directly from the Outer zone.

4.2.4 Delay - Model 5 and Model 6

Similar to Model4, almost all rooms in Model5 are accessed from the outside. The exception being Model5's central room, which obviously must be reached via a neighbour. To force the system to produce a more interesting result where an attacker has to weave through the building you require something like Model6.

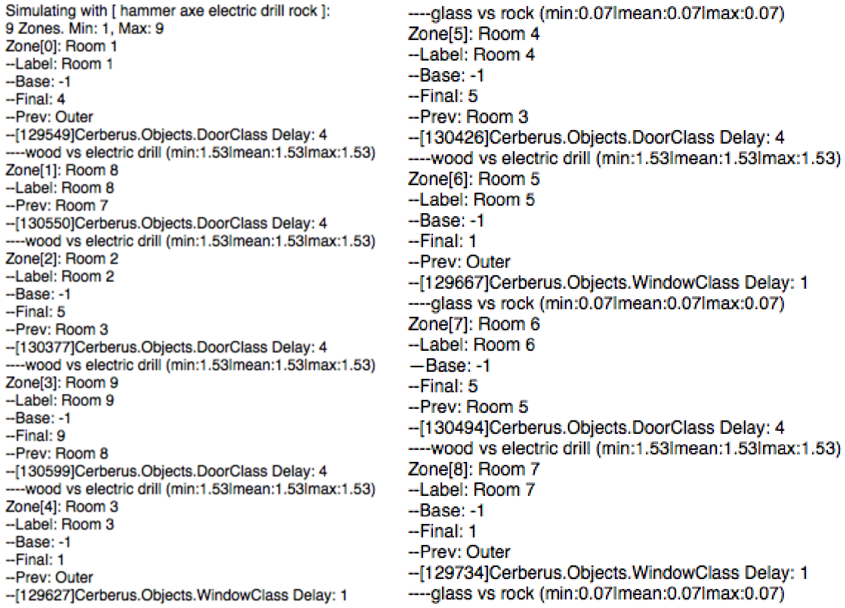
Looking at Model6 and using our knowledge of how Model4 operated we can predict that the system will not find a long, snaking path through the facility. The windows in Room 3, Room 5 and Room 7 provide cheap access to those rooms and the rooms beside them.



```
Simulating with [ hammer axe electric drill rock ]:
4 Zones. Min: 1, Max: 4
Outer:
--Label: Outer
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Prev: Outer
--[129549]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[1]: Room 4
--Label: Room 4
--Prev: Outer
--[129734]Cerberus.Objects.WindowClass Delay: 1
----glass vs rock (min:0.07|mean:0.07|max:0.07)
Zone[2]: Room 2
--Label: Room 2
--Prev: Outer
--[129627]Cerberus.Objects.WindowClass Delay: 1
----glass vs rock (min:0.07|mean:0.07|max:0.07)
Zone[3]: Room 3
--Label: Room 3
--Prev: Outer
--[129667]Cerberus.Objects.WindowClass Delay: 1
----glass vs rock (min:0.07|mean:0.07|max:0.07)
```

Figure 4.19: An excerpt from the log of Model4's security simulation, edited down to only the points of interest

The results in Figure 4.20 play this out.



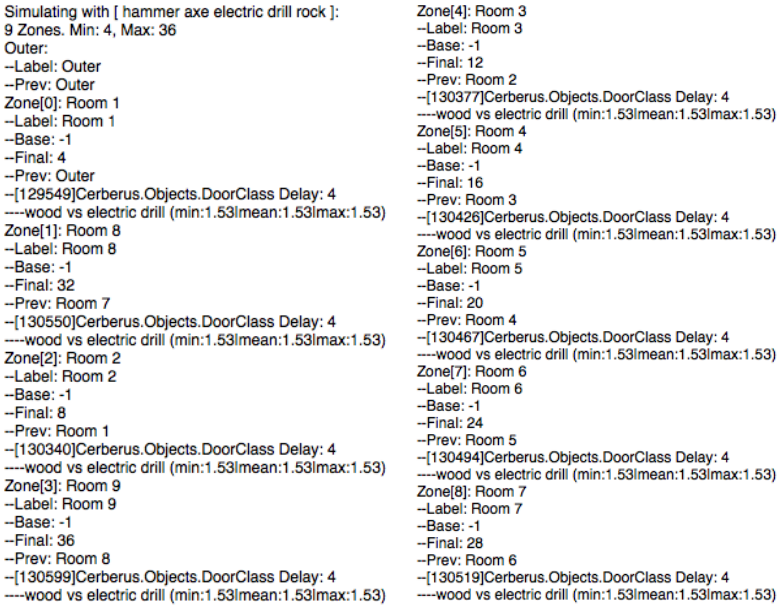
```
Simulating with [ hammer axe electric drill rock ]:
9 Zones. Min: 1, Max: 9
Zone[0]: Room 1
--Label: Room 1
--Base: -1
--Final: 4
--Prev: Outer
--[129549]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[1]: Room 8
--Label: Room 8
--Prev: Room 7
--[130550]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[2]: Room 2
--Label: Room 2
--Base: -1
--Final: 5
--Prev: Room 3
--[130377]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[3]: Room 9
--Label: Room 9
--Base: -1
--Final: 9
--Prev: Room 8
--[130599]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[4]: Room 3
--Label: Room 3
--Base: -1
--Final: 1
--Prev: Outer
--[129627]Cerberus.Objects.WindowClass Delay: 1
----glass vs rock (min:0.07|mean:0.07|max:0.07)
Zone[5]: Room 4
--Label: Room 4
--Base: -1
--Final: 5
--Prev: Room 3
--[130426]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[6]: Room 5
--Label: Room 5
--Base: -1
--Final: 1
--Prev: Outer
--[129667]Cerberus.Objects.WindowClass Delay: 1
----glass vs rock (min:0.07|mean:0.07|max:0.07)
Zone[7]: Room 6
--Label: Room 6
—Base: -1
--Final: 5
--Prev: Room 5
--[130494]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[8]: Room 7
--Label: Room 7
--Base: -1
--Final: 1
--Prev: Outer
--[129734]Cerberus.Objects.WindowClass Delay: 1
----glass vs rock (min:0.07|mean:0.07|max:0.07)
```

Figure 4.20: An excerpt from the log of Model6's security simulation, edited down to only the points of interest

If we look at the "-Final:" field on the entries we can see the final delay values, i.e. the minimum cost to reach that room via available paths. We can see that the value is low on rooms such as Room 3 (Final: 1) which can be externally accessed via windows. The highest value can be seen on Room 9 (Final: 9), which being centrally located requires a window and two doors to be breached before it can be accessed.

4.2.5 Delay - Model 7

Model7 is designed entirely to force the long chain of previous rooms that did not occur in the other models. It is analogous with having a goal delay or a goal path you would wish to force attackers along that causes them to penetrate areas with sensors attached and so have hardened all other paths. The results can be seen in Figure 4.21



```
Simulating with [ hammer axe electric drill rock ]:
9 Zones. Min: 4, Max: 36
Outer:
--Label: Outer
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Base: -1
--Final: 4
--Prev: Outer
--[129549]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[1]: Room 8
--Label: Room 8
--Base: -1
--Final: 32
--Prev: Room 7
--[130550]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[2]: Room 2
--Label: Room 2
--Base: -1
--Final: 8
--Prev: Room 1
--[130340]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[3]: Room 9
--Label: Room 9
--Base: -1
--Final: 36
--Prev: Room 8
--[130599]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[4]: Room 3
--Label: Room 3
--Base: -1
--Final: 12
--Prev: Room 2
--[130377]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[5]: Room 4
--Label: Room 4
--Base: -1
--Final: 16
--Prev: Room 3
--[130426]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[6]: Room 5
--Label: Room 5
--Base: -1
--Final: 20
--Prev: Room 4
--[130467]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[7]: Room 6
--Label: Room 6
--Base: -1
--Final: 24
--Prev: Room 5
--[130494]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
Zone[8]: Room 7
--Label: Room 7
--Base: -1
--Final: 28
--Prev: Room 6
--[130519]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
```

Figure 4.21: An excerpt from the log of Model7's security log

We can see from the feedback that the maximum delay is only 36 seconds, but that is still considerably higher than the maximum delay of 9 seconds we saw in Model6. Tracing the previous zone links, we can also see the attacker is forced to trace a path through the entire facility as we had hoped. If we considered Model7 an attempt to harden Model6 and make Room 9 difficult to access then we would appear to have succeeded.

Looking at the feedback from Model7 we can see that each zone is accessed from the previous zone by breaching the door. If we wished to harden this model further, we could experiment with altering those doors and see how it affects the delays for each room. At present it is necessary for a user to access the security log to see such details as the previous chain, but the system results shown in Figure 4.15 will also allow a user to explore changes to the facility, though in less granularity.

4.2.6 Delay - Height

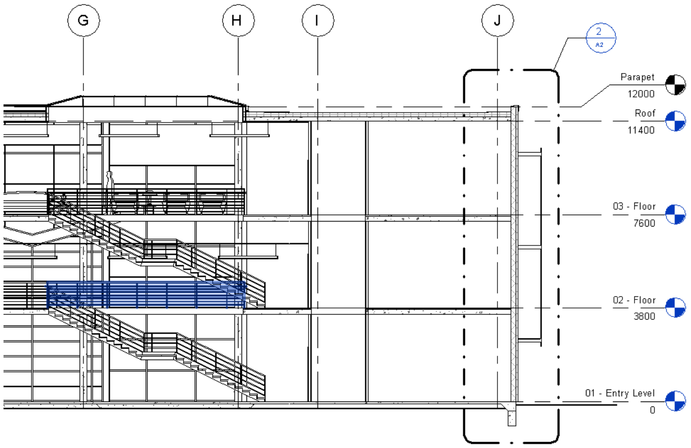
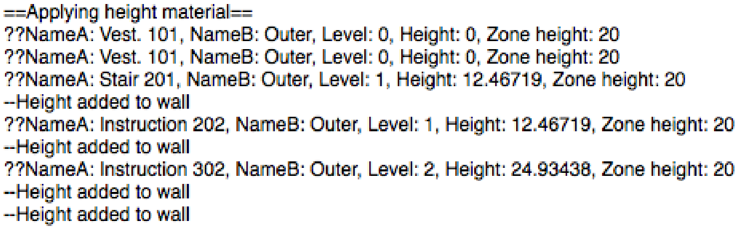


Figure 4.22: The RAC_Advanced_Sample_Project model included with Revit® Architecture 2012

As discussed previously, our own models do not seem to contain the requisite height information to allow proper testing of the height material. Instead we rely upon our modified version of the RAC_Advanced_Sample_Project, a cutaway of which can be seen in Figure 4.22. The model is considerably more complex than the ones we have dealt with above but our system is capable of processing it.

If we look to the log of the graph creation, we can see that the system first detects the levels and then at the end of the graph generation assigns the height materials. An excerpt of the log can be seen in Figure 4.23. We can see that the first few items examined, Vest. 101s barriers, have a height of 0 and are not assigned any height material, as we would expect.



```
==Applying height material==
??NameA: Vest. 101, NameB: Outer, Level: 0, Height: 0, Zone height: 20
??NameA: Vest. 101, NameB: Outer, Level: 0, Height: 0, Zone height: 20
??NameA: Stair 201, NameB: Outer, Level: 1, Height: 12.46719, Zone height: 20
--Height added to wall
??NameA: Instruction 202, NameB: Outer, Level: 1, Height: 12.46719, Zone height: 20
--Height added to wall
??NameA: Instruction 302, NameB: Outer, Level: 2, Height: 24.93438, Zone height: 20
--Height added to wall
--Height added to wall
```

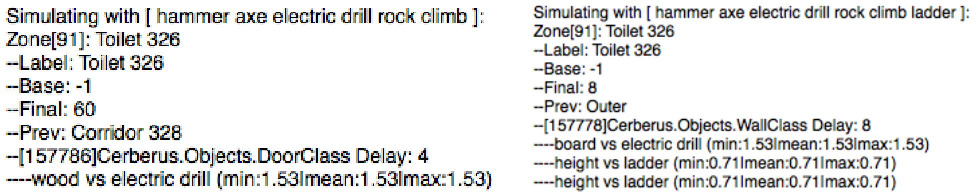
Figure 4.23: A small excerpt from the run log of RAC_Advanced_Sample_Project

If we look at Stair 201 we can tell from the 201 and its height that it lies on the second floor. We can also see on the line below it a log reference, "–Height added to wall" indicating that the system has attached height material to that barrier. This is as we would expect and allows us to see that the system is working.

The final entry is for Instruction 302, indicating it belongs to a room on the third level. We can see the height value is roughly twice the height of the second floor room we examined. We can also see the layer system at work, applying two measures of height material to the wall, simulating two levels of height needing to be overcome.

If we look at the security report for the model, seen in Figure 4.24, we can see the same room compared across two runs of the system with different tools available. We can see from the final delay values that the tool selections make quite a difference and the previous zone along with the ingress method gives us an indication as to why. In the later run on the right, the quickest path is to access the zone from the outer zone with the use of the Ladder.

The room we have selected is from the third floor so we know that exterior facing barriers should have two layers of height added. We can see this in the right hand excerpt, with the ladder used twice to overcome height before the electric drill is used to breach the board. It is interesting to note that in the left hand simulation results, exterior access is possible via the climb attack, but the delay cost of doing so is higher than accessing via a path within the facility.



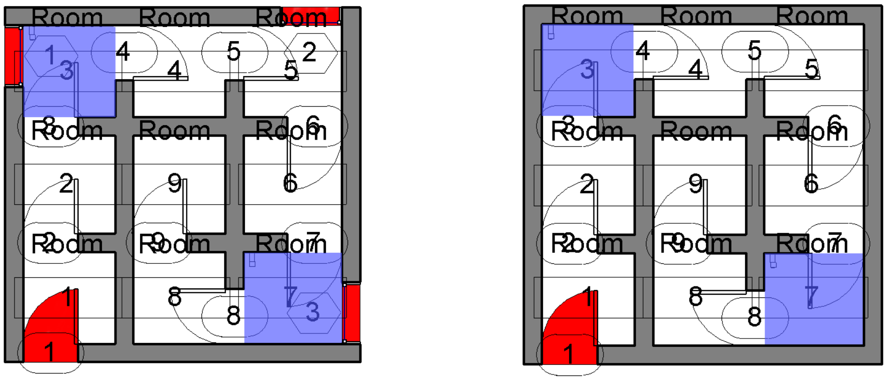
```
Simulating with [ hammer axe electric drill rock climb ]:
Zone[91]: Toilet 326
--Label: Toilet 326
--Base: -1
--Final: 60
--Prev: Corridor 328
--[157786]Cerberus.Objects.DoorClass Delay: 4
----wood vs electric drill (min:1.53|mean:1.53|max:1.53)
```

```
Simulating with [ hammer axe electric drill rock climb ladder ]:
Zone[91]: Toilet 326
--Label: Toilet 326
--Base: -1
--Final: 8
--Prev: Outer
--[157778]Cerberus.Objects.WallClass Delay: 8
----board vs electric drill (min:1.53|mean:1.53|max:1.53)
----height vs ladder (min:0.71|mean:0.71|max:0.71)
----height vs ladder (min:0.71|mean:0.71|max:0.71)
```

Figure 4.24: A side by side comparison of Toilet 326 with different tool sets

The above results support the fact that our system can simulate both larger, more complex facilities and facilities with multiple levels. Simulating a complex, multi-level facility does take some time, approximately 58 seconds for the above described model to complete its run with 6 tools. However, given this is the result when running under a virtualised copy of Windows running on a fairly average iMac and we have put minimal effort into code optimisation we think this is quite acceptable and should scale well to larger projects. The feedback may not be instant but it is timely for any designer who wishes to use it.

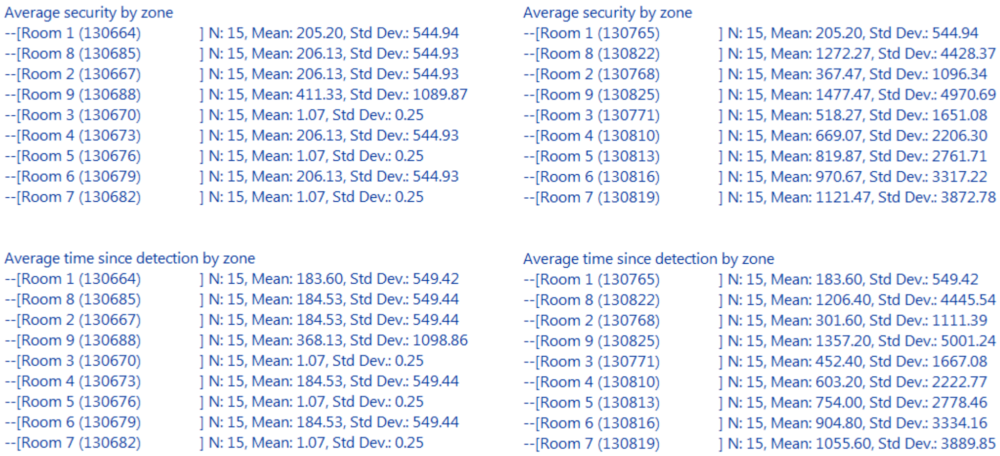
4.2.7 Detect & Detain



(a) Model6 (b) Model7

Figure 4.25: Model6 beside Model7 with detection regions highlighted

To test detection, we modified Model6 and Model7 to add reed sensors and video cameras within the BIM. In Figure 4.25, we have highlighted the doors and windows we altered with red and coloured the rooms featuring cameras with blue. In the case of the modified Model6, we do not expect any of the Time Since Delay (TSD) values to be particularly high because of the numerous entry points, but Model7 should feature an ever increasing TSD inline with its delay values.



```
Average security by zone
--[Room 1 (130664)          ] N: 15, Mean: 205.20, Std Dev.: 544.94
--[Room 8 (130685)          ] N: 15, Mean: 206.13, Std Dev.: 544.93
--[Room 2 (130667)          ] N: 15, Mean: 206.13, Std Dev.: 544.93
--[Room 9 (130688)          ] N: 15, Mean: 411.33, Std Dev.: 1089.87
--[Room 3 (130670)          ] N: 15, Mean: 1.07, Std Dev.: 0.25
--[Room 4 (130673)          ] N: 15, Mean: 206.13, Std Dev.: 544.93
--[Room 5 (130676)          ] N: 15, Mean: 1.07, Std Dev.: 0.25
--[Room 6 (130679)          ] N: 15, Mean: 206.13, Std Dev.: 544.93
--[Room 7 (130682)          ] N: 15, Mean: 1.07, Std Dev.: 0.25

Average time since detection by zone
--[Room 1 (130664)          ] N: 15, Mean: 183.60, Std Dev.: 549.42
--[Room 8 (130685)          ] N: 15, Mean: 184.53, Std Dev.: 549.44
--[Room 2 (130667)          ] N: 15, Mean: 184.53, Std Dev.: 549.44
--[Room 9 (130688)          ] N: 15, Mean: 368.13, Std Dev.: 1098.86
--[Room 3 (130670)          ] N: 15, Mean: 1.07, Std Dev.: 0.25
--[Room 4 (130673)          ] N: 15, Mean: 184.53, Std Dev.: 549.44
--[Room 5 (130676)          ] N: 15, Mean: 1.07, Std Dev.: 0.25
--[Room 6 (130679)          ] N: 15, Mean: 184.53, Std Dev.: 549.44
--[Room 7 (130682)          ] N: 15, Mean: 1.07, Std Dev.: 0.25
```

(a) Model6

```
Average security by zone
--[Room 1 (130765)          ] N: 15, Mean: 205.20, Std Dev.: 544.94
--[Room 8 (130822)          ] N: 15, Mean: 1272.27, Std Dev.: 4428.37
--[Room 2 (130768)          ] N: 15, Mean: 367.47, Std Dev.: 1096.34
--[Room 9 (130825)          ] N: 15, Mean: 1477.47, Std Dev.: 4970.69
--[Room 3 (130771)          ] N: 15, Mean: 518.27, Std Dev.: 1651.08
--[Room 4 (130810)          ] N: 15, Mean: 669.07, Std Dev.: 2206.30
--[Room 5 (130813)          ] N: 15, Mean: 819.87, Std Dev.: 2761.71
--[Room 6 (130816)          ] N: 15, Mean: 970.67, Std Dev.: 3317.22
--[Room 7 (130819)          ] N: 15, Mean: 1121.47, Std Dev.: 3872.78

Average time since detection by zone
--[Room 1 (130765)          ] N: 15, Mean: 183.60, Std Dev.: 549.42
--[Room 8 (130822)          ] N: 15, Mean: 1206.40, Std Dev.: 4445.54
--[Room 2 (130768)          ] N: 15, Mean: 301.60, Std Dev.: 1111.39
--[Room 9 (130825)          ] N: 15, Mean: 1357.20, Std Dev.: 5001.24
--[Room 3 (130771)          ] N: 15, Mean: 452.40, Std Dev.: 1667.08
--[Room 4 (130810)          ] N: 15, Mean: 603.20, Std Dev.: 2222.77
--[Room 5 (130813)          ] N: 15, Mean: 754.00, Std Dev.: 2778.46
--[Room 6 (130816)          ] N: 15, Mean: 904.80, Std Dev.: 3334.16
--[Room 7 (130819)          ] N: 15, Mean: 1055.60, Std Dev.: 3889.85
```

(b) Model7

Figure 4.26: Excerpts of the security reports for Model6 and Model7

The results, seen in Figure 4.26 match these expectations, though they can be slightly hard to trace as printed. To help clarify, in order of access the Model7 TSD read as:

- Room 1 - 183.60
- Room 2 - 301.60
- Room 3 - 454.40
- Room 4 - 603.20
- Room 5 - 754.00
- Room 6 - 904.80
- Room 7 - 1055.60
- Room 8 - 1206.40
- Room 9 - 1357.20

The report for Model6 can be confusing upon first read and indeed it even caused us to go back and double check our figures. The figures are correct as printed and the confusion can be put down to the problems of paths and the weakness of certain materials. Room 3, 5 and 7 all average very low values because they feature windows and glass is very quick to breach regardless of what tools are available.

The TSD value for Room 1 is nearly the same as for Room 2s, which seems counter intuitive. The door for Room 2 cannot be that much quicker to breach than the door for Room 1 with the same tools, raising the question of how they so quickly accessed Room 2 from Room 1. The trick is that Room 2 is not accessed via Room 1 as we might expect, it is accessed via Room 3.

Thus the TSD of Room 2 is the TSD of Room 3 plus the delay cost of breaching the door between the two. The TSD of Room 1 is the cost of breaching its door, resulting in the two values being nearly identical because of the ease of breaching a window. Once a user has these values they can investigate them as we did to determine their causes and then take appropriate steps, such as hardening the window if necessary.

Given the above information a user could determine through other tests that the average Response Force Time is 5 minutes. Looking at the above values, they could determine that a Response force is likely to intervene before an attacker reaches Room 9 on Model6. For any other room some kind of modification is required if detainment is important for that room.

4.3 Summary

In this chapter we have described the design and implementation of our security simulation aspects. We have detailed the process we undertook to convert the data harvested from a BIM into a model suitable for security simulation and highlighted current areas of weakness. As part of that process we highlighted areas of difficulty, such as height, and our solutions to them.

We detailed the objects used to capture the facility model and the methods used by them to perform the security simulation. We then demonstrated and explored the process of security simulation to show that it works as expected. We also repeated the process against a variety of models of increasing complexity to show our system's ability to handle realistic designs. Finally, we demonstrated the functionality of our system's detection mechanic.

In the next and final chapter, we shall discuss the implications of our work to date. We will summarise our results and examine them in comparison to our goals. Finally, we will look at ways in which our work might be extended in the future.

Chapter 5

Conclusion and Discussion

In the previous chapters we have discussed research within the domains of Building Information Models, Security and Security Simulation as well as our own work. In this chapter we will discuss the results of our work with reference to our initial aims. We will also go on to discuss the conclusions that can be drawn from our work and some of the directions future work could take.

5.1 Discussion

In this section we will discuss our results with reference to our original aims. We will then discuss ethical considerations arising from our work.

5.1.1 Aims and Results

In the introduction chapter we discussed four aims for our research. We will present each below and discuss the results of each stage of our research.

Research and Design
We aim to first identify appropriate toolsets and methods for the development of the system. We will explore existing related systems and develop the design of our own. This will include work on deciding a appropriate level of abstraction for our security models.

Within chapter 2, Background, we examined existing works that deal with the exploitation of information within BIMs as well as work related to computerised security aids and security methodologies. Based on the works we examined, BIMs provide a great deal of potential to support novel tasks that would be impossible or impractical under traditional facility modelling. Likewise there has been a good deal of promising and interesting work on security simulation and automation.

From the research examined, little within the field of Building Information Modelling was directly applicable, though the work by Grason (1971) provided some similarities that helped to advance our thinking. Within security simulation the work by Tarr and Peaty (1995), Alach (2007) and Garcia (2001) were the most informative and instructive towards developing our own system. These works and the others discussed provided a strong theoretical foundation for us to build upon.

From research of the available toolsets, as covered in chapter 3 on Building Information Modelling, we found that commercial toolsets were most appropriate for our work. From our examination they provided the best support and would allow for the easiest collection of existing models. The support for modifying software through extensions would provide the quickest route to being able to make use of the data contained within a BIM.

We have used the above information to design and develop a system which strikes an acceptable balance between abstraction and functionality. As such, we have fulfilled our aim of gathering sufficient information to support the design of a system for security analysis of building information models.

Identify, Extract and Model Data

Secondly, we will aim to identify data available with Building Information Models that is useful for security modelling. We will explore and develop methods of extracting the necessary data and compiling it in meaningful and accessible formats.

Identification of data for modelling was achieved by comparison of the data available within BIMs to the data used in vulnerability assessment. Early on, possible routes and materials were the aspects of interest as they encapsulate barriers and their delay factors. As discussed in the chapter on BIM we were able to, through extensive research combined with trial and error, identify the storage methods and encoding for these parameters within our chosen BIM suite.

We then developed an object oriented dataset to capture this information and allow its modelling in a graph format, such that we might then leverage known graph solving algorithms such as Dijkstra's shortest path algorithm against it. This allowed us to model the data at a high fidelity with the addition of information in our classes to allow reference back to the original BIM elements. We have achieved this goal to a high degree of satisfaction and have provided details within this thesis to help others emulate our methods.

Development of Automated Vulnerability Assessment
Thirdly, we aim to explore and develop a system of utilising data extracted from BIM to provide Automated Vulnerability Assessment. We intend for this to demonstrate the viability of incorporating security domain knowledge into a process that can produce useful feedback for less knowledgeable users.

In chapter 4, Security, we give details on our process for simulating security and the output generated by our system to assist the user in making decisions. We have elected for a quantitative analysis and have examined the output within the chapter and discussed its interpretation. As a proof of concept, we have shown that the system can perform complex analysis and provide data that would be useful in facility hardening and softening.

In the future, the quantity, type of feedback and its presentation might be improved through consultation with potential users. We believe the information we generate is of interest to designers, but its current form may be somewhat unintuitive. There may also be information we do not currently model that might be of interest to potential users.

We feel we have met this aim with the use of our proof of concept system, although further work is possible to improve the system we developed.

Evaluate
The final aim will be to evaluate our methods and provide analysis of their success.

Evaluations in previous chapters demonstrate the strong potential of Building Information Modelling, Computerised Security Simulation and the interaction of the two. We believe our research to be a strong success with potential for expansion through future research.

5.1.2 Ethical Consideration

During our research we have considered the ramification of software designed to find security weaknesses within a facility that is useable by even a relatively untrained user. Obviously an attacker with access to our system and the relevant data could potentially use it to map a facility to find their best points of ingress and egress. This could put certain assets at increased risk if our system was used to circumvent intended delay, detect and detain strategies.

In our opinion this risk is negligible however, as to achieve such an analysis an attacker would already need access to a detailed and up to date BIM of the facility including security arrangements. As such data is generally well protected due to commercial sensitivity concerns, it is unlikely an attacker would have access to it. In the event that an attacker does have access to the relevant data, without our software they would still have enough sensitive details to formulate their own attack plan.

While our system might make formulating an attack easier, we feel any risks posed by this are outweighed by the potential benefits of early stage design hardening. Building Information Models are already treated with care in industry, so the security of the facility would no doubt be proportional to the difficulty of acquiring its BIM. Based on this, existing industry policies governing access to BIMs should be sufficient to ensure that our system does not introduce a greatly increased security risk.

5.2 Conclusion

This research set out to explore the potential of Building Information Models to support a security simulation toolset. We explored the BIM knowledge base and discovered not only that BIM had a great deal of potential, but also several interesting projects have already taken advantage of it. Having established to our satisfaction that BIMs would provide a suitable information basis for our security simulation, we conducted research into our platform options.

We selected the best platform for our purposes, Autodesk® Revit® , and undertook the process of extracting data contained within BIMs. We found that the process of accessing the data we wanted was more difficult than we had expected, with useful data sometimes stored in buried properties. This difficulty informed our decision to include a great deal of data on our process within our chapter on BIM, to assist others attempting similar tasks.

Once we could reliably access useful data, we began mapping it to a suitable graph abstraction we had formulated. Applying security simulation concepts based on prior research and our own reading, we implemented a proof of concept system able to handle realistically large and complex facilities. We have documented within this thesis the results generated by our system to demonstrate its functionality and capability.

Like most research, our work has been more of a journey than a destination. The work to date has suggested ways in which our system could be extended and improved to further leverage the potential of BIMs and computer simulation. We feel this research has real potential to assist designers and security practitioners in hardening facilities and producing secure designs cheaply.

The research has been challenging and rewarding. BIMs and Security Simulation are two quite disparate fields, so there has been little support available for many problems unique to the integration of the two. Examining these two fields from the point of view of an outsider has greatly expanded my knowledge of areas that are normally considered separate to computer science, providing a rewarding challenge.

As BIM moves towards being the standard for the Architecture, Engineering and Construction industry, the potential for research like ours to value add will increase considerably. Better adoption of standards, particularly for information sharing, will assist the industry in maximising their return on investment and allow for even greater value add to their clients. We hope this research will assist others working towards this goal as well as acting as a stand alone demonstration of the potential of Building Information Modelling and computer aided vulnerability assessment.

5.3 Future Work

As with any research there is more we would have liked to have seen done but did not have the resources to achieve at this time. In this section we will detail some of the ways in which our research could be extended. Each of these avenues could provide useful dividends, though some will present better returns than others.

5.3.1 User Accessible Interface

The introduction of an interface for the easy selection and setting of criteria for the simulation would provide the most immediate benefit in making our system more accessible. The current reliance on a programmer to alter code and then recompile the software before exploring other tool sets presents a clear limitation to the use of our system. As this was a research exercise, the penalty to date has been minimal, but we would like to see the system extended for easier access and implementation.

The system would also most likely benefit from a wider range of output, the type and style of which could be established through industry consultation. More integrated feedback may also improve usability, such as adding 3D objects back into the BIM that can be clicked to establish relative security and risks of an area visually. We did examine the possibility of adding such objects during this research but the difficulty to implement made it impossible to complete within our timeline.

As we saw in Tarr and Peaty (1995), a scalable user friendly interface can greatly improve the uptake of a system and improve its saturation within an organisation. Were our system to be adopted as a commercial prospect, an improved interface for input and output would greatly assist it in aiding users to improve designs.

5.3.2 Human Traversal Delays

To improve the fidelity of the simulation, we have considered the addition of variable delays for attackers when moving across a zone. These delays can be undertaken with varying degrees of complexity, with the simplest method being to simply apply an arbitrary cost by measuring the straight line distance between the points of ingress and egress. Such a method fails to consider complexities such as strangely shaped rooms and corridors or furniture interfering with the direct path, but provides an adequate estimate for some uses.

The more realistic, and therefore complex method, would be to analyse the internal geometry within a zone. You could then generate a map and plan a path allowing for furniture and obstructing walls. We have seen work similar to this in the paper by Lee *et al.* (2010).

In the paper they introduce the Universal Circulation Network (UCN), a system which creates more human-like paths within modelled environments. Their modelling takes account of physical limitations, such as doors, as well as psychological, such as personal space. It results in paths that are closer to the path a human would choose, as can be seen in Figure 5.1 diagram f.

Such a method could be further extended to try and capture elements like attacker fitness and carrying weight. These elements could provide interesting elements of realism to the simulation, requiring attackers to offload and carefully select equipment for an attack. Such extensions would provide only limited returns in terms of vulnerability assessment but might present interesting research opportunities.

Improved path finding would assist our system in performing realistic simulations. It would also open up the opportunity for the system to be extended to simulate emergency

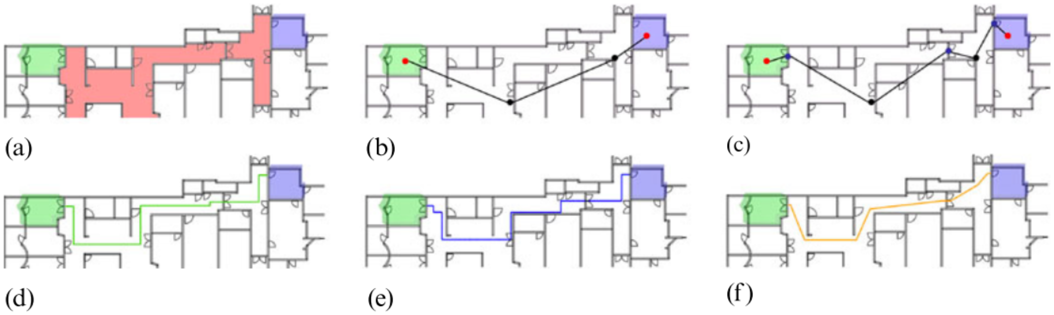


Figure 5.1: A copy of Figure 1 from Lee et al. *(2010) with example navigational graphs, diagram f being a representation of UCN.*

situations such as evacuations due to fire or attack. Tools for emergency simulations already exist, so that aspect would mainly be beneficial as a further value-add.

5.3.3 GIS Integration

Geographical Information Systems (GIS) contain information which can impact security considerations, such as possible access routes via roads or underground tunnels as well as power and communications infrastructure. Integrating such information into security simulations would enable a broader view, allowing facilities to better prepare for high level orchestrated attacks. The implementation of such work may only benefit a small subset of high value targets, but may be justified for implementing in the future.

Information on roads would provide the most obvious benefits and could potentially be paired with some form of tracking to allow for the system to be used in a control room situation. For instance, if a facility registers an alarm, a real time map on the location of all response units could help determine which ones are best located to intervene. This may also allow an operator to de-emphasise attacks that cannot be interrupted in time, allowing resources to be directed to where they will have a more positive effect.

Sewage routes and power supply modelling would likely have a much more limited utility. While Hollywood movies may portray terrorists and prisoners as utilising utility pipes, real world incidents are considerably less common. For high value targets, simulation and consideration of such possibilities may be worth while, but the effort required to implement these features for a subset of users may not be worthwhile when they should already have the resources for a thorough security analysis.

5.3.4 Intelligent Agent Based Simulation

Intelligent Agents could be added to our existing simulation to take on various roles, such as Designer, Attacker and Defender. For instance, a Designer agent could take a set of design parameters, such as security equipment cost, and run simulations with variants of equipment type and placement to help find hardening solutions for a facility. Defender and Attacker agents could be used to perform simulated red teaming, adapting to each other and experimenting with various attack and defence layouts to help a designer find the best compromises between personnel and patrol paths.

With the aid of a colleague, Terence Tan, we have already explored some aspects of this approach. In relatively little time Terence was able to take advantage of our XML output and a Java based program of his own design to simulate red teaming against the provided facility graph. A paper detailing our work, Porter *et al.* (2014), has been published in the journal Automation In Construction (Impact Factor 1.82).

The simulated red teaming experimented with a “red” agent attacking a facility, followed by a “blue” agent examining their method of attack and then hardening the facility to try and slow or dissuade attack by that avenue. His software also created false colour heat maps of facilities, as can be seen in Figure 5.2, highlighting the vulnerabilities of various rooms. The work to date has been fairly limited and there is ample scope for further research along this avenue.

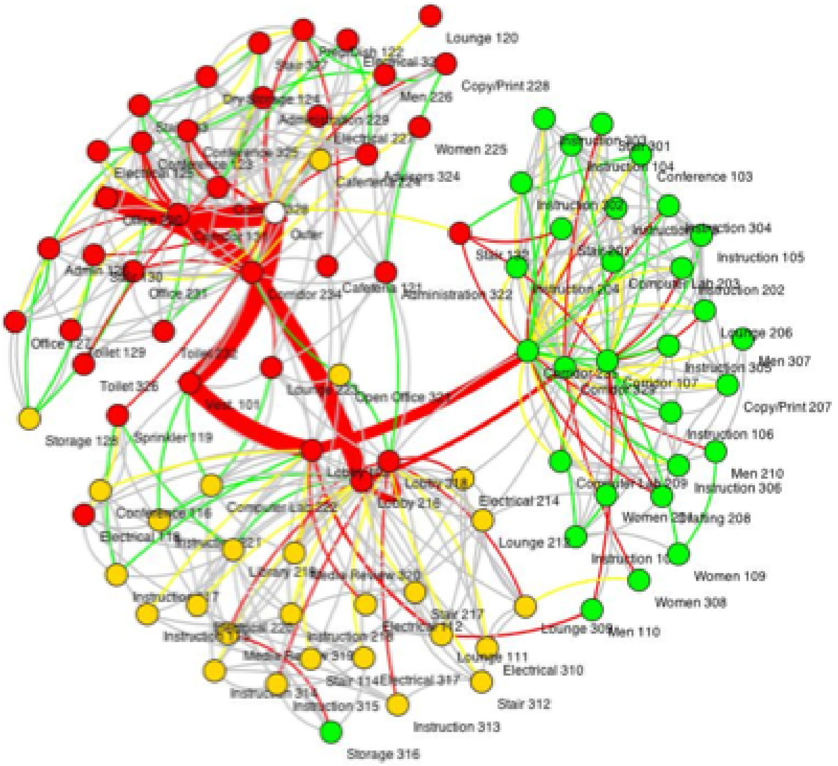


Figure 5.2: A heatmap of generated by Tan's graph software with nodes labelled for their respective rooms. Nodes are colour coded according to relative security, Red indicating low while Green indicates high. Extracted from Porter et al. (2014)

All three aspects of the proposed agents offer interesting potential. If sufficiently developed, the designer agent could greatly assist in facility design, one of the primary uses we envision for our software. Based on the results we've seen so far with only limited investment, any development towards interaction between our work and intelligent agents is likely to pay useful dividends.

5.3.5 Multi-Factor Simulation

Multi-factor simulation would look at elements an attacker might consider besides the most expedient path. Equipment cost or skill might be of interest, although they can already be manually explored to some degree via the existing equipment system. Stealth would be the element we consider most of interest for future development.

Windows are typically used as an attack path under our current modelling because glass is so quickly penetrated, but breaking windows creates noise and leaves an obvious breach point which may be noticed, due to the broken glass. It would be interesting to investigate the construction of a model that balances an attacker's desire to avoid drawing attention with their desire to quickly access their target. Such a model might be further extended by attempting to capture behaviour an attacker might use to avoid cameras and other detection methods.

We believe this avenue for future research would likely provide the best benefits, assuming appropriate ways to model attacker decision making could be formulated and implemented. The ability for a designer to explore and understand how an attacker might approach their facility could provide an immense advantage in designing the facility to resist attacks. Such a problem is highly complex however, as it can involve fuzzy elements such as human psychology, but the rewards could be great if it can be solved.

5.3.6 Simulations, Estimations and Optimisations

As has been covered, the information layer in BIMs provides a great deal of potential for simulation and estimation. In conjunction with our system's ability to extract and model information, many opportunities exist beyond those presented above. The limitations are the level of information available, the amount required for a given simulation and whether it can be extracted by the processes used in our system.

An example is carbon footprint modelling. Databases exist which contain the carbon impact of various materials used in construction. Using these databases in conjunction with BIM could allow interested parties to estimate the volume of materials required and their carbon impact, allowing them to compare the impacts of varying materials.

Such a system could be extended with the addition of intelligent agents, allowing companies to automate the process of finding a solution to carbon reduction targets. These technologies could similarly assist with other simulations and optimisations, such as heat and energy load. In this way the system we've created could assist in many areas of facility design, augmenting BIM's existing capabilities to help improve efficiency across a range of fields.

Bibliography

Administration, U. G. S. (2013). P-100 Facilities Standards for the Public Buildings Service. `www.gsa.gov/p100`. 16

Alach, Z. J. (2007). *Mapping the elements of physical security towards the creation of a holistic physical security model*. Master of Science (Security Science) thesis, Edith Cowan University. vi, 27, 29, 70, 71, 80, 84, 94

Azhar, S., Hein, M., and Sketo, B. (2008). Building Information Modeling (BIM): Benefits , Risks and Challenges. In *Proceedings of the 44th ASC Annual Conference (on CD ROM)*, Auburn, Alabama. vi, 2, 12, 13

Boeing, A., Masek, M., and Bailey, B. (2003). Protecting Critical Infrastructure with Games Technology. In *Proceedings of the 4th Australian Information Warfare and Security Conference*, Perth, Western Australia. SECAU. 2

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. 2, 8, 9

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271. 65

Eastman, C. (2009). Automated Assessment of Early Concept Designs. *Architectural Design*, **79**(2), 52–57. vi, 16, 18

Eastman, C. M., Fisher, D., Lafue, G., Lividini, J., Stoker, D., and Yessios, C. (1974). An outline of the building description system. Technical Report 50, Institute of Physical Planning, Carnegie-Mellon University, Pittsburgh, PA. 7

Gao, J. and Fischer, M. (2008). *Framework and Case Studies Comparing Implementations and Impacts of 3D/4D Modeling Across Projects*. Ph.D. (Civil and Environmental Engineering) thesis, Stanford University. vi, 10, 11

Garcia, M. L. (2001). *The Design and Evaluation of Physical Protection Systems*, chapter 14 - Easi Computer Model for Analysis. Butterworth-Heinemann. 2, 25, 64, 94

Garcia, M. L. (2005). *Vulnerability Assessment of Physical Protection Systems*. Butterworth-Heinemann. 26

Global Terrorism Database (2012). National Consortium for the Study of Terorism and Response to Terrorism (START). 1

Grason, J. (1971). An Approach to Computerized Space Planning Using Graph Theory. In *Workshop on Design Automation*, pages 170–179. vi, 13, 14, 15, 94

Lee, J.-k., Eastman, C. M., Lee, J.-m., Kannala, M., and Jeong, Y.-s. (2010). Computing walking distances within buildings using the universal circulation network. *Environment and Planning. B, Planning and Design*, **37**(4), 628–645. viii, 98, 99

Osman, E. A., El-Gazar, M. I., Shaat, M. K., El-Kafas, A. A., Zidan, W. I., and A., W. A. (2009). An Estimation Of A Passive Infra-Red Sensor's Probability Of Detection. In *Proceedings of the 7th Conference on Nuclear and Particle Physics*, pages 493–502, Sharam El-Sheikh, Egypt. 76

Porter, S., Tan, T., Tan, T., and West, G. (2014). Breaking into BIM: Performing static and dynamic security analysis with the aid of BIM. *Automation in Construction*, **40**, 84 – 95. viii, 76, 100, 101

Smith, D. K. (2009). Ensuring building performance through simulation. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 2672–2678. Ieee. 9

Tarr, C. J. (1992). CLASP: a computerised aid to cost effective perimeter security. In *Proceedings of IEEE International Carnahan Conference on Security Technology*, pages 164–168. Ieee. vi, 19, 20, 21, 64

Tarr, C. J. (1994). Cost effective perimeter security. In *Proceedings of IEEE International Carnahan Conference on Security Technology*, pages 60–65. Ieee. vi, 21, 22, 64

Tarr, C. J. and Peaty, S. (1995). Using CLASP to assess perimeter security. In *Proceedings of IEEE International Carnahan Conference on Security Technology*, pages 311–316. Ieee. vi, 2, 23, 24, 64, 94, 98

Ustun, V., Yilmaz, L., and Smith, J. S. (2006). A conceptual model for agent-based simulation of physical security systems. In *Proceedings of the 44th annual southeast regional conference on - ACM-SE 44*, page 365, New York, New York, USA. ACM Press. 2

Vance, R. (2013). US Courts Design Guide. www.wbdg.org/ccb/GSAMAN/courts.pdf. 16

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Appendix A

Demonstration model graph as XML: Version 1 through 7

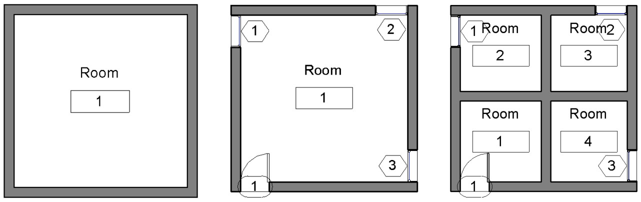


Figure A.1: V1state, V2state & V3state model

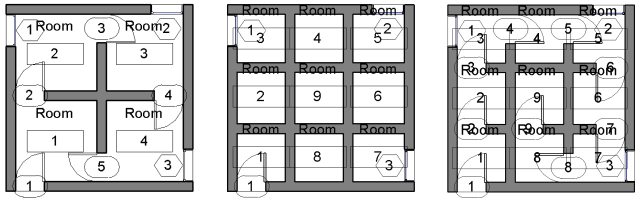


Figure A.2: V4state, V5state & V6state model

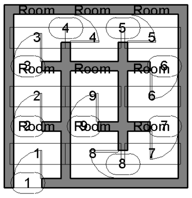


Figure A.3: V7state model

```xml
<state>
  <minSecurity>56</minSecurity>
  <maxSecurity>56</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>56</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
      </walls>
      <doors>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone0>
  </zones>
</state>
```

```
<state>
  <minSecurity>1</minSecurity>
  <maxSecurity>1</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Outer</neighbour>
        </door0>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
        <window1>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
```

```
          </window1>
          <window2>
            <security>1</security>
            <material>glass</material>
            <neighbour>Outer</neighbour>
          </window2>
        </windows>
        <conceptuals>
        </conceptuals>
        <travels>
        </travels>
        <roofs>
        </roofs>
        <floors>
        </floors>
        <environments>
        </environments>
      </zone0>
    </zones>
  </state>
```

```
<state>
  <minSecurity>1</minSecurity>
  <maxSecurity>4</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>4</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 4</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Outer</neighbour>
        </door0>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
```

```
        <roofs>
        </roofs>
        <floors>
        </floors>
        <environments>
        </environments>
      </zone0>
      <zone1>
        <name>Room 4</name>
        <initialSecurity>-1</initialSecurity>
        <currentSecurity>1</currentSecurity>
        <previous>Outer</previous>
        <walls>
          <wall0>
            <security>56</security>
            <material>brick</material>
            <neighbour>Room 1</neighbour>
          </wall0>
          <wall1>
            <security>56</security>
            <material>brick</material>
            <neighbour>Room 3</neighbour>
          </wall1>
          <wall2>
            <security>56</security>
            <material>brick</material>
            <neighbour>Outer</neighbour>
          </wall2>
          <wall3>
            <security>56</security>
            <material>brick</material>
            <neighbour>Outer</neighbour>
          </wall3>
        </walls>
        <doors>
        </doors>
        <windows>
          <window0>
            <security>1</security>
            <material>glass</material>
            <neighbour>Outer</neighbour>
          </window0>
        </windows>
        <conceptuals>
        </conceptuals>
        <travels>
        </travels>
        <roofs>
        </roofs>
        <floors>
        </floors>
        <environments>
        </environments>
      </zone1>
      <zone2>
        <name>Room 2</name>
        <initialSecurity>-1</initialSecurity>
        <currentSecurity>1</currentSecurity>
        <previous>Outer</previous>
        <walls>
          <wall0>
            <security>56</security>
            <material>brick</material>
            <neighbour>Room 1</neighbour>
          </wall0>
```

```xml
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
      </walls>
      <doors>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone2>
    <zone3>
      <name>Room 3</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 4</neighbour>
        </wall3>
      </walls>
      <doors>
      </doors>
```

```xml
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone3>
  </zones>
</state>
```

```xml
<state>
  <minSecurity>1</minSecurity>
  <maxSecurity>4</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>4</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 4</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Outer</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 2</neighbour>
        </door1>
        <door2>
          <security>4</security>
```

```
          <material>wood</material>
          <neighbour>Room 4</neighbour>
        </door2>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone0>
    <zone1>
      <name>Room 4</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 1</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 3</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 1</neighbour>
        </door1>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
```

```
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone1>
    <zone2>
      <name>Room 2</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 1</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 1</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 3</neighbour>
        </door1>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
```

```xml
    </zone2>
    <zone3>
      <name>Room 3</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 4</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 2</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 4</neighbour>
        </door1>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone3>
  </zones>
</state>
```

```
<state>
  <minSecurity>1</minSecurity>
  <maxSecurity>112</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>4</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 8</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Outer</neighbour>
        </door0>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
```

```
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone0>
    <zone1>
      <name>Room 8</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>56</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 1</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 7</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall3>
      </walls>
      <doors>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone1>
    <zone2>
      <name>Room 2</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>56</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 1</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
```

```
            <wall2>
              <security>56</security>
              <material>brick</material>
              <neighbour>Room 9</neighbour>
            </wall2>
            <wall3>
              <security>56</security>
              <material>brick</material>
              <neighbour>Room 3</neighbour>
            </wall3>
          </walls>
          <doors>
          </doors>
          <windows>
          </windows>
          <conceptuals>
          </conceptuals>
          <travels>
          </travels>
          <roofs>
          </roofs>
          <floors>
          </floors>
          <environments>
          </environments>
        </zone2>
        <zone3>
          <name>Room 9</name>
          <initialSecurity>-1</initialSecurity>
          <currentSecurity>112</currentSecurity>
          <previous>Room 8</previous>
          <walls>
            <wall0>
              <security>56</security>
              <material>brick</material>
              <neighbour>Room 2</neighbour>
            </wall0>
            <wall1>
              <security>56</security>
              <material>brick</material>
              <neighbour>Room 4</neighbour>
            </wall1>
            <wall2>
              <security>56</security>
              <material>brick</material>
              <neighbour>Room 6</neighbour>
            </wall2>
            <wall3>
              <security>56</security>
              <material>brick</material>
              <neighbour>Room 8</neighbour>
            </wall3>
          </walls>
          <doors>
          </doors>
          <windows>
          </windows>
          <conceptuals>
          </conceptuals>
          <travels>
          </travels>
          <roofs>
          </roofs>
          <floors>
          </floors>
```

```
      <environments>
      </environments>
    </zone3>
    <zone4>
      <name>Room 3</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 4</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
      </walls>
      <doors>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone4>
    <zone5>
      <name>Room 4</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>56</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
```

```
          </wall1>
          <wall2>
            <security>56</security>
            <material>brick</material>
            <neighbour>Room 9</neighbour>
          </wall2>
          <wall3>
            <security>56</security>
            <material>brick</material>
            <neighbour>Room 5</neighbour>
          </wall3>
        </walls>
        <doors>
        </doors>
        <windows>
        </windows>
        <conceptuals>
        </conceptuals>
        <travels>
        </travels>
        <roofs>
        </roofs>
        <floors>
        </floors>
        <environments>
        </environments>
      </zone5>
      <zone6>
        <name>Room 5</name>
        <initialSecurity>-1</initialSecurity>
        <currentSecurity>1</currentSecurity>
        <previous>Outer</previous>
        <walls>
          <wall0>
            <security>56</security>
            <material>brick</material>
            <neighbour>Room 4</neighbour>
          </wall0>
          <wall1>
            <security>56</security>
            <material>brick</material>
            <neighbour>Outer</neighbour>
          </wall1>
          <wall2>
            <security>56</security>
            <material>brick</material>
            <neighbour>Outer</neighbour>
          </wall2>
          <wall3>
            <security>56</security>
            <material>brick</material>
            <neighbour>Room 6</neighbour>
          </wall3>
        </walls>
        <doors>
        </doors>
        <windows>
          <window0>
            <security>1</security>
            <material>glass</material>
            <neighbour>Outer</neighbour>
          </window0>
        </windows>
        <conceptuals>
        </conceptuals>
```

```xml
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone6>
    <zone7>
      <name>Room 6</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>56</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 5</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 7</neighbour>
        </wall3>
      </walls>
      <doors>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone7>
    <zone8>
      <name>Room 7</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 6</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
```

```
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 8</neighbour>
        </wall3>
      </walls>
      <doors>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone8>
  </zones>
</state>
```

```xml
<state>
  <minSecurity>1</minSecurity>
  <maxSecurity>9</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>4</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 8</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Outer</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 2</neighbour>
        </door1>
      </doors>
      <windows>
```

```
    </windows>
    <conceptuals>
    </conceptuals>
    <travels>
    </travels>
    <roofs>
    </roofs>
    <floors>
    </floors>
    <environments>
    </environments>
  </zone0>
  <zone1>
    <name>Room 8</name>
    <initialSecurity>-1</initialSecurity>
    <currentSecurity>5</currentSecurity>
    <previous>Room 7</previous>
    <walls>
      <wall0>
        <security>56</security>
        <material>brick</material>
        <neighbour>Room 1</neighbour>
      </wall0>
      <wall1>
        <security>56</security>
        <material>brick</material>
        <neighbour>Room 7</neighbour>
      </wall1>
      <wall2>
        <security>56</security>
        <material>brick</material>
        <neighbour>Outer</neighbour>
      </wall2>
      <wall3>
        <security>56</security>
        <material>brick</material>
        <neighbour>Room 9</neighbour>
      </wall3>
    </walls>
    <doors>
      <door0>
        <security>4</security>
        <material>wood</material>
        <neighbour>Room 7</neighbour>
      </door0>
      <door1>
        <security>4</security>
        <material>wood</material>
        <neighbour>Room 9</neighbour>
      </door1>
    </doors>
    <windows>
    </windows>
    <conceptuals>
    </conceptuals>
    <travels>
    </travels>
    <roofs>
    </roofs>
    <floors>
    </floors>
    <environments>
    </environments>
  </zone1>
  <zone2>
```

```xml
      <name>Room 2</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>5</currentSecurity>
      <previous>Room 3</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 1</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 1</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 3</neighbour>
        </door1>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone2>
    <zone3>
      <name>Room 9</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>9</currentSecurity>
      <previous>Room 8</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
```

```xml
          <neighbour>Room 4</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 6</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 8</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 8</neighbour>
        </door0>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone3>
    <zone4>
      <name>Room 3</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 4</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 2</neighbour>
```

```xml
                </door0>
                <door1>
                  <security>4</security>
                  <material>wood</material>
                  <neighbour>Room 4</neighbour>
                </door1>
              </doors>
              <windows>
                <window0>
                  <security>1</security>
                  <material>glass</material>
                  <neighbour>Outer</neighbour>
                </window0>
              </windows>
              <conceptuals>
              </conceptuals>
              <travels>
              </travels>
              <roofs>
              </roofs>
              <floors>
              </floors>
              <environments>
              </environments>
            </zone4>
            <zone5>
              <name>Room 4</name>
              <initialSecurity>-1</initialSecurity>
              <currentSecurity>5</currentSecurity>
              <previous>Room 3</previous>
              <walls>
                <wall0>
                  <security>56</security>
                  <material>brick</material>
                  <neighbour>Room 3</neighbour>
                </wall0>
                <wall1>
                  <security>56</security>
                  <material>brick</material>
                  <neighbour>Outer</neighbour>
                </wall1>
                <wall2>
                  <security>56</security>
                  <material>brick</material>
                  <neighbour>Room 9</neighbour>
                </wall2>
                <wall3>
                  <security>56</security>
                  <material>brick</material>
                  <neighbour>Room 5</neighbour>
                </wall3>
              </walls>
              <doors>
                <door0>
                  <security>4</security>
                  <material>wood</material>
                  <neighbour>Room 3</neighbour>
                </door0>
                <door1>
                  <security>4</security>
                  <material>wood</material>
                  <neighbour>Room 5</neighbour>
                </door1>
              </doors>
              <windows>
```

```
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone5>
    <zone6>
      <name>Room 5</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 4</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 6</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 4</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 6</neighbour>
        </door1>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
```

```
      </floors>
      <environments>
      </environments>
    </zone6>
    <zone7>
      <name>Room 6</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>5</currentSecurity>
      <previous>Room 5</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 5</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 7</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 5</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 7</neighbour>
        </door1>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone7>
    <zone8>
      <name>Room 7</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
```

```
          <neighbour>Room 6</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 8</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 6</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 8</neighbour>
        </door1>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone8>
  </zones>
</state>
```

```xml
<state>
  <minSecurity>4</minSecurity>
  <maxSecurity>36</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>4</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 8</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Outer</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 2</neighbour>
        </door1>
      </doors>
      <windows>
```

```
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone0>
    <zone1>
      <name>Room 8</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>32</currentSecurity>
      <previous>Room 7</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 1</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 7</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 7</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 9</neighbour>
        </door1>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone1>
    <zone2>
```

```
      <name>Room 2</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>8</currentSecurity>
      <previous>Room 1</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 1</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 1</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 3</neighbour>
        </door1>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone2>
    <zone3>
      <name>Room 9</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>36</currentSecurity>
      <previous>Room 8</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 2</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
```

```
      <neighbour>Room 4</neighbour>
    </wall1>
    <wall2>
      <security>56</security>
      <material>brick</material>
      <neighbour>Room 6</neighbour>
    </wall2>
    <wall3>
      <security>56</security>
      <material>brick</material>
      <neighbour>Room 8</neighbour>
    </wall3>
  </walls>
  <doors>
    <door0>
      <security>4</security>
      <material>wood</material>
      <neighbour>Room 8</neighbour>
    </door0>
  </doors>
  <windows>
  </windows>
  <conceptuals>
  </conceptuals>
  <travels>
  </travels>
  <roofs>
  </roofs>
  <floors>
  </floors>
  <environments>
  </environments>
</zone3>
<zone4>
  <name>Room 3</name>
  <initialSecurity>-1</initialSecurity>
  <currentSecurity>12</currentSecurity>
  <previous>Room 2</previous>
  <walls>
    <wall0>
      <security>56</security>
      <material>brick</material>
      <neighbour>Room 2</neighbour>
    </wall0>
    <wall1>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall1>
    <wall2>
      <security>56</security>
      <material>brick</material>
      <neighbour>Room 4</neighbour>
    </wall2>
    <wall3>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall3>
  </walls>
  <doors>
    <door0>
      <security>4</security>
      <material>wood</material>
      <neighbour>Room 2</neighbour>
```

```
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 4</neighbour>
        </door1>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone4>
    <zone5>
      <name>Room 4</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>16</currentSecurity>
      <previous>Room 3</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 5</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 3</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 5</neighbour>
        </door1>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
```

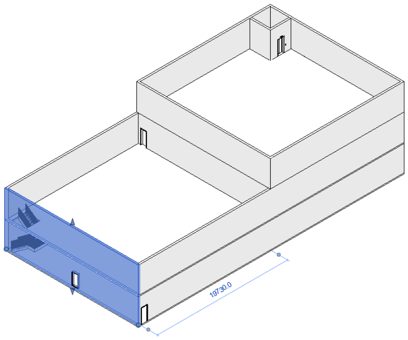
```
  <roofs>
  </roofs>
  <floors>
  </floors>
  <environments>
  </environments>
</zone5>
<zone6>
  <name>Room 5</name>
  <initialSecurity>-1</initialSecurity>
  <currentSecurity>20</currentSecurity>
  <previous>Room 4</previous>
  <walls>
    <wall0>
      <security>56</security>
      <material>brick</material>
      <neighbour>Room 4</neighbour>
    </wall0>
    <wall1>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall1>
    <wall2>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall2>
    <wall3>
      <security>56</security>
      <material>brick</material>
      <neighbour>Room 6</neighbour>
    </wall3>
  </walls>
  <doors>
    <door0>
      <security>4</security>
      <material>wood</material>
      <neighbour>Room 4</neighbour>
    </door0>
    <door1>
      <security>4</security>
      <material>wood</material>
      <neighbour>Room 6</neighbour>
    </door1>
  </doors>
  <windows>
  </windows>
  <conceptuals>
  </conceptuals>
  <travels>
  </travels>
  <roofs>
  </roofs>
  <floors>
  </floors>
  <environments>
  </environments>
</zone6>
<zone7>
  <name>Room 6</name>
  <initialSecurity>-1</initialSecurity>
  <currentSecurity>24</currentSecurity>
  <previous>Room 5</previous>
  <walls>
```

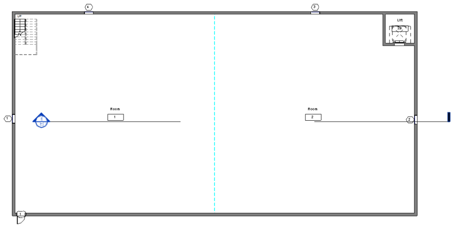
```
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 5</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 9</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 7</neighbour>
        </wall3>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 5</neighbour>
        </door0>
        <door1>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 7</neighbour>
        </door1>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
      </travels>
      <roofs>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone7>
    <zone8>
      <name>Room 7</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>28</currentSecurity>
      <previous>Room 6</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 6</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
```

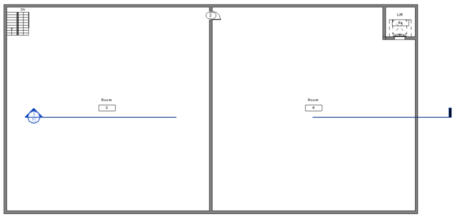
```
        <neighbour>Outer</neighbour>
      </wall2>
      <wall3>
        <security>56</security>
        <material>brick</material>
        <neighbour>Room 8</neighbour>
      </wall3>
    </walls>
    <doors>
      <door0>
        <security>4</security>
        <material>wood</material>
        <neighbour>Room 6</neighbour>
      </door0>
      <door1>
        <security>4</security>
        <material>wood</material>
        <neighbour>Room 8</neighbour>
      </door1>
    </doors>
    <windows>
    </windows>
    <conceptuals>
    </conceptuals>
    <travels>
    </travels>
    <roofs>
    </roofs>
    <floors>
    </floors>
    <environments>
    </environments>
  </zone8>
 </zones>
</state>
```

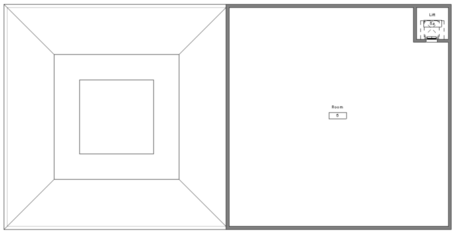
Appendix B

Multi-level model graph as XML









```xml
<state>
  <minSecurity>1</minSecurity>
  <maxSecurity>1</maxSecurity>
  <tools>
    <tool0>
      <name>hammer</name>
    </tool0>
    <tool1>
      <name>axe</name>
    </tool1>
    <tool2>
      <name>electric drill</name>
    </tool2>
    <tool3>
      <name>rock</name>
    </tool3>
    <tool4>
      <name>climb</name>
    </tool4>
    <tool5>
      <name>ladder</name>
    </tool5>
  </tools>
  <zones>
    <zone0>
      <name>Room 1</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
      </walls>
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Outer</neighbour>
        </door0>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
        <window1>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window1>
```

```xml
      </windows>
      <conceptuals>
        <conceptual0>
          <security>0</security>
          <neighbour>Room 2</neighbour>
        </conceptual0>
      </conceptuals>
      <travels>
        <travel0>
          <security>0</security>
          <neighbour>Room 3</neighbour>
        </travel0>
      </travels>
      <roofs>
        <roof0>
          <security>0</security>
          <neighbour>Room 3</neighbour>
        </roof0>
      </roofs>
      <floors>
      </floors>
      <environments>
      </environments>
    </zone0>
    <zone1>
      <name>Room 2</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Outer</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall3>
        <wall4>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall4>
      </walls>
      <doors>
      </doors>
      <windows>
        <window0>
          <security>1</security>
          <material>glass</material>
          <neighbour>Outer</neighbour>
        </window0>
        <window1>
```

```xml
      <security>1</security>
      <material>glass</material>
      <neighbour>Outer</neighbour>
    </window1>
  </windows>
  <conceptuals>
    <conceptual0>
      <security>0</security>
      <neighbour>Room 1</neighbour>
    </conceptual0>
  </conceptuals>
  <travels>
    <travel0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </travel0>
    <travel1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </travel1>
  </travels>
  <roofs>
    <roof0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </roof0>
    <roof1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </roof1>
  </roofs>
  <floors>
  </floors>
  <environments>
  </environments>
</zone1>
<zone2>
  <name>Room 3</name>
  <initialSecurity>-1</initialSecurity>
  <currentSecurity>1</currentSecurity>
  <previous>Room 1</previous>
  <walls>
    <wall0>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall0>
    <wall1>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall1>
    <wall2>
      <security>56</security>
      <material>brick</material>
      <neighbour>Room 4</neighbour>
    </wall2>
    <wall3>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall3>
  </walls>
  <doors>
    <door0>
```

```xml
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 4</neighbour>
        </door0>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
        <travel0>
          <security>0</security>
          <neighbour>Room 1</neighbour>
        </travel0>
      </travels>
      <roofs>
      </roofs>
      <floors>
        <floor0>
          <security>0</security>
          <neighbour>Room 1</neighbour>
        </floor0>
      </floors>
      <environments>
      </environments>
    </zone2>
    <zone3>
      <name>Room 4</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Room 2</previous>
      <walls>
        <wall0>
          <security>56</security>
          <material>brick</material>
          <neighbour>Room 3</neighbour>
        </wall0>
        <wall1>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall1>
        <wall2>
          <security>58</security>
          <material>brick</material>
          <material>height</material>
          <neighbour>Outer</neighbour>
        </wall2>
        <wall3>
          <security>58</security>
          <material>brick</material>
          <material>height</material>
          <neighbour>Outer</neighbour>
        </wall3>
        <wall4>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall4>
        <wall5>
          <security>56</security>
          <material>brick</material>
          <neighbour>Outer</neighbour>
        </wall5>
      </walls>
```

```
      <doors>
        <door0>
          <security>4</security>
          <material>wood</material>
          <neighbour>Room 3</neighbour>
        </door0>
      </doors>
      <windows>
      </windows>
      <conceptuals>
      </conceptuals>
      <travels>
        <travel0>
          <security>0</security>
          <neighbour>Room 2</neighbour>
        </travel0>
        <travel1>
          <security>0</security>
          <neighbour>Room 5</neighbour>
        </travel1>
        <travel2>
          <security>0</security>
          <neighbour>Lift 5a</neighbour>
        </travel2>
        <travel3>
          <security>0</security>
          <neighbour>Lift 2a</neighbour>
        </travel3>
      </travels>
      <roofs>
        <roof0>
          <security>0</security>
          <neighbour>Room 5</neighbour>
        </roof0>
        <roof1>
          <security>0</security>
          <neighbour>Lift 5a</neighbour>
        </roof1>
      </roofs>
      <floors>
        <floor0>
          <security>0</security>
          <neighbour>Room 2</neighbour>
        </floor0>
        <floor1>
          <security>0</security>
          <neighbour>Lift 2a</neighbour>
        </floor1>
      </floors>
      <environments>
      </environments>
    </zone3>
    <zone4>
      <name>Room 5</name>
      <initialSecurity>-1</initialSecurity>
      <currentSecurity>1</currentSecurity>
      <previous>Room 4</previous>
      <walls>
        <wall0>
          <security>60</security>
          <material>brick</material>
          <material>height</material>
          <material>height</material>
          <neighbour>Outer</neighbour>
        </wall0>
```

```xml
    <wall1>
      <security>60</security>
      <material>brick</material>
      <material>height</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall1>
    <wall2>
      <security>60</security>
      <material>brick</material>
      <material>height</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall2>
    <wall3>
      <security>58</security>
      <material>brick</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall3>
    <wall4>
      <security>58</security>
      <material>brick</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall4>
    <wall5>
      <security>60</security>
      <material>brick</material>
      <material>height</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall5>
  </walls>
  <doors>
  </doors>
  <windows>
  </windows>
  <conceptuals>
  </conceptuals>
  <travels>
    <travel0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </travel0>
    <travel1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </travel1>
  </travels>
  <roofs>
  </roofs>
  <floors>
    <floor0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </floor0>
    <floor1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </floor1>
  </floors>
  <environments>
  </environments>
</zone4>
```

```
<zone5>
  <name>Lift 5a</name>
  <initialSecurity>-1</initialSecurity>
  <currentSecurity>1</currentSecurity>
  <previous>Room 4</previous>
  <walls>
    <wall0>
      <security>60</security>
      <material>brick</material>
      <material>height</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall0>
    <wall1>
      <security>60</security>
      <material>brick</material>
      <material>height</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall1>
    <wall2>
      <security>58</security>
      <material>brick</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall2>
    <wall3>
      <security>58</security>
      <material>brick</material>
      <material>height</material>
      <neighbour>Outer</neighbour>
    </wall3>
  </walls>
  <doors>
  </doors>
  <windows>
  </windows>
  <conceptuals>
  </conceptuals>
  <travels>
    <travel0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </travel0>
    <travel1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </travel1>
  </travels>
  <roofs>
  </roofs>
  <floors>
    <floor0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </floor0>
    <floor1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </floor1>
  </floors>
  <environments>
  </environments>
</zone5>
<zone6>
```

```xml
<name>Lift 4a</name>
<initialSecurity>-1</initialSecurity>
<currentSecurity>1</currentSecurity>
<previous>Room 2</previous>
<walls>
  <wall0>
    <security>56</security>
    <material>brick</material>
    <neighbour>Outer</neighbour>
  </wall0>
  <wall1>
    <security>56</security>
    <material>brick</material>
    <neighbour>Outer</neighbour>
  </wall1>
  <wall2>
    <security>58</security>
    <material>brick</material>
    <material>height</material>
    <neighbour>Outer</neighbour>
  </wall2>
  <wall3>
    <security>58</security>
    <material>brick</material>
    <material>height</material>
    <neighbour>Outer</neighbour>
  </wall3>
</walls>
<doors>
</doors>
<windows>
</windows>
<conceptuals>
</conceptuals>
<travels>
  <travel0>
    <security>0</security>
    <neighbour>Room 2</neighbour>
  </travel0>
  <travel1>
    <security>0</security>
    <neighbour>Room 5</neighbour>
  </travel1>
  <travel2>
    <security>0</security>
    <neighbour>Lift 5a</neighbour>
  </travel2>
  <travel3>
    <security>0</security>
    <neighbour>Lift 2a</neighbour>
  </travel3>
</travels>
<roofs>
  <roof0>
    <security>0</security>
    <neighbour>Room 5</neighbour>
  </roof0>
  <roof1>
    <security>0</security>
    <neighbour>Lift 5a</neighbour>
  </roof1>
</roofs>
<floors>
  <floor0>
    <security>0</security>
```

```xml
      <neighbour>Room 2</neighbour>
    </floor0>
    <floor1>
      <security>0</security>
      <neighbour>Lift 2a</neighbour>
    </floor1>
  </floors>
  <environments>
  </environments>
</zone6>
<zone7>
  <name>Lift 2a</name>
  <initialSecurity>-1</initialSecurity>
  <currentSecurity>1</currentSecurity>
  <previous>Room 4</previous>
  <walls>
    <wall0>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall0>
    <wall1>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall1>
    <wall2>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall2>
    <wall3>
      <security>56</security>
      <material>brick</material>
      <neighbour>Outer</neighbour>
    </wall3>
  </walls>
  <doors>
  </doors>
  <windows>
  </windows>
  <conceptuals>
  </conceptuals>
  <travels>
    <travel0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </travel0>
    <travel1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </travel1>
  </travels>
  <roofs>
    <roof0>
      <security>0</security>
      <neighbour>Room 4</neighbour>
    </roof0>
    <roof1>
      <security>0</security>
      <neighbour>Lift 4a</neighbour>
    </roof1>
  </roofs>
  <floors>
  </floors>
```

```xml
      <environments>
      </environments>
    </zone7>
  </zones>
</state>
```

Appendix C

A Materials XML input file

Stored as "materials.xml" and read in at runtime.
The names of possible attacks are listed in the header section so that the relevant objects may be created for use in the materials that are created in the main body.

```
<Cerberus>
        <Attacks>
                <Attack>
                        <Name>rock</Name>
                </Attack>
                <Attack>
                        <Name>hammer</Name>
                </Attack>
                <Attack>
                        <Name>axe</Name>
                </Attack>
                <Attack>
                        <Name>oxy cutting</Name>
                </Attack>
                <Attack>
                        <Name>electric drill</Name>
                </Attack>
                <Attack>
                        <Name>explosive</Name>
                </Attack>
                <Attack>
                        <Name>climb</Name>
                </Attack>
                <Attack>
                        <Name>ladder</Name>
                </Attack>
        </Attacks>

        <Materials>
                <Material>
                        <Name>glass</Name>
                        <Attacks>
                                <Attack>
                                <Name>rock</Name>
                                <Delay>0.07</Delay>
                                </Attack>

                                <Attack>
                                <Name>hammer</Name>
                                <Delay>0.24</Delay>
                                </Attack>

                                <Attack>
                                <Name>axe</Name>
                                <Delay>0.07</Delay>
                                </Attack>

                                <Attack>
                                <Name>oxy cutting</Name>
                                <Delay>1.68</Delay>
                                </Attack>

                                <Attack>
                                <Name>electric drill</Name>
```

```
                <Delay>0.71</Delay>
                </Attack>

                <Attack>
                <Name>explosive</Name>
                <Delay>0.07</Delay>
                </Attack>
        </Attacks>
</Material>

<Material>
        <Name>earth</Name>
        <Attacks>
                <Attack>
                <Name>rock</Name>
                <Delay>0.76</Delay>
                </Attack>

                <Attack>
                <Name>hammer</Name>
                <Delay>0.95</Delay>
                </Attack>

                <Attack>
                <Name>axe</Name>
                <Delay>1.08</Delay>
                </Attack>

                <Attack>
                <Name>oxy cutting</Name>
                <Delay>1.68</Delay>
                </Attack>

                <Attack>
                <Name>electric drill</Name>
                <Delay>0.55</Delay>
                </Attack>

                <Attack>
                <Name>explosive</Name>
                <Delay>0.23</Delay>
                </Attack>
        </Attacks>
</Material>

<Material>
        <Name>wood</Name>
        <Attacks>
                <Attack>
                <Name>rock</Name>
                <Delay>7.71</Delay>
                </Attack>

                <Attack>
```

```
                                <Name>hammer</Name>
                                <Delay>4.82</Delay>
                                </Attack>

                                <Attack>
                                <Name>axe</Name>
                                <Delay>5.93</Delay>
                                </Attack>

                                <Attack>
                                <Name>oxy cutting</Name>
                                <Delay>3.15</Delay>
                                </Attack>

                                <Attack>
                                <Name>electric drill</Name>
                                <Delay>1.53</Delay>
                                </Attack>

                                <Attack>
                                <Name>explosive</Name>
                                <Delay>0.07</Delay>
                                </Attack>
                        </Attacks>
                </Material>

                <Material>
                        <Name>board</Name>
                        <Attacks>
                                <Attack>
                                <Name>rock</Name>
                                <Delay>7.71</Delay>
                                </Attack>

                                <Attack>
                                <Name>hammer</Name>
                                <Delay>4.82</Delay>
                                </Attack>

                                <Attack>
                                <Name>axe</Name>
                                <Delay>5.93</Delay>
                                </Attack>

                                <Attack>
                                <Name>oxy cutting</Name>
                                <Delay>3.15</Delay>
                                </Attack>

                                <Attack>
                                <Name>electric drill</Name>
                                <Delay>1.53</Delay>
                                </Attack>
```

```
                <Attack>
                <Name>explosive</Name>
                <Delay>0.07</Delay>
                </Attack>
        </Attacks>
</Material>

<Material>
        <Name>brick</Name>
        <Attacks>
                <Attack>
                <Name>rock</Name>
                <Delay>10.85</Delay>
                </Attack>

                <Attack>
                <Name>hammer</Name>
                <Delay>5.14</Delay>
                </Attack>

                <Attack>
                <Name>axe</Name>
                <Delay>5.09</Delay>
                </Attack>

                <Attack>
                <Name>oxy cutting</Name>
                <Delay>8.51</Delay>
                </Attack>

                <Attack>
                <Name>electric drill</Name>
                <Delay>4.04</Delay>
                </Attack>

                <Attack>
                <Name>explosive</Name>
                <Delay>0.23</Delay>
                </Attack>
        </Attacks>
</Material>

<Material>
        <Name>concrete</Name>
        <Attacks>
                <Attack>
                <Name>rock</Name>
                <Delay>14.51</Delay>
                </Attack>

                <Attack>
                <Name>hammer</Name>
                <Delay>8.59</Delay>
                </Attack>
```

```
                                <Attack>
                                <Name>axe</Name>
                                <Delay>8.07</Delay>
                                </Attack>

                                <Attack>
                                <Name>oxy cutting</Name>
                                <Delay>9.59</Delay>
                                </Attack>

                                <Attack>
                                <Name>electric drill</Name>
                                <Delay>5.39</Delay>
                                </Attack>

                                <Attack>
                                <Name>explosive</Name>
                                <Delay>0.39</Delay>
                                </Attack>
                        </Attacks>
                </Material>

                <Material>
                        <Name>steel</Name>
                        <Attacks>
                                <Attack>
                                <Name>rock</Name>
                                <Delay>17.30</Delay>
                                </Attack>

                                <Attack>
                                <Name>hammer</Name>
                                <Delay>13.53</Delay>
                                </Attack>

                                <Attack>
                                <Name>axe</Name>
                                <Delay>12.22</Delay>
                                </Attack>

                                <Attack>
                                <Name>oxy cutting</Name>
                                <Delay>3.71</Delay>
                                </Attack>

                                <Attack>
                                <Name>electric drill</Name>
                                <Delay>6.63</Delay>
                                </Attack>

                                <Attack>
                                <Name>explosive</Name>
                                <Delay>1.04</Delay>
```

```
                                </Attack>
                        </Attacks>
                </Material>
                <Material>
                        <Name>height</Name>
                        <Attacks>
                                <Attack>
                                <Name>climb</Name>
                                <Delay>12.22</Delay>
                                </Attack>

                                <Attack>
                                <Name>ladder</Name>
                                <Delay>0.71</Delay>
                                </Attack>
                        </Attacks>
                </Material>

        </Materials>
</Cerberus>
```

Appendix D

Model1 run log

A copy of the output generated during graph generation. We can see the system move through the layers of materials on each barrier, indicating those it can and cannot match.

```
==[Cerberus beginning run @ 1/08/2013 4:05:48 PM]==
==Establishing levels==
Level Ground Floor(311) is at 0
Level Level 1(694) is at 13.1233595800525
==Graphing: Wall and seperation==

*Room 1 boundary segments*
-Room 1:Double brick - 270
>>>Could not match: Air Barrier - Air Infiltration Barrier
>>>Could not match: Masonry
!!!Material assigned: Masonry - Brick - Brown
--Linked 129488:129320(WALL):-1
-Room 1:Double brick - 270
>>>Could not match: Air Barrier - Air Infiltration Barrier
>>>Could not match: Masonry
!!!Material assigned: Masonry - Brick - Brown
--Linked 129488:129362(WALL):-1
-Room 1:Double brick - 270
>>>Could not match: Air Barrier - Air Infiltration Barrier
>>>Could not match: Masonry
!!!Material assigned: Masonry - Brick - Brown
--Linked 129488:129393(WALL):-1
-Room 1:Double brick - 270
>>>Could not match: Air Barrier - Air Infiltration Barrier
>>>Could not match: Masonry
!!!Material assigned: Masonry - Brick - Brown
--Linked 129488:129413(WALL):-1
Floor is at level 0
Stairs(28598) 190mm max riser 250mm going @ level N/A
Stairs(28599) Industrial and Assembly @ level N/A
Stairs(28600) Part M (Disabled) @ level N/A
Stairs(28601) Private @ level N/A
Stairs(55159) Monolithic Stair @ level N/A
==SPECIALTY EQUIPMENT==
==Applying height material==
??NameA: Room 1, NameB: Outer, Level: 0, Height: 0, Zone height: 20
??NameA: Room 1, NameB: Outer, Level: 0, Height: 0, Zone height: 20
??NameA: Room 1, NameB: Outer, Level: 0, Height: 0, Zone height: 20
??NameA: Room 1, NameB: Outer, Level: 0, Height: 0, Zone height: 20
==[Cerberus ending run @ 1/08/2013 4:05:50 PM]==
```

Appendix E

Model1 security log

The complete log of the simulation run against Model1 against all permutations of the tools selected.

```
==[Cerberus beginning run @ 1/08/2013 4:05:48 PM]==
Simulating with [ hammer ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)
Simulating with [ hammer axe ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ hammer axe electric drill ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer axe electric drill rock ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer axe electric drill rock climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer axe electric drill rock climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
```

```
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ hammer axe electric drill rock ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ hammer axe electric drill climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer axe electric drill climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
```

```
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)



Simulating with [ hammer axe electric drill ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)



Simulating with [ hammer axe rock ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ hammer axe rock climb ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ hammer axe rock climb ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)




Simulating with [ hammer axe rock ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
```

```
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)




Simulating with [ hammer axe climb ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ hammer axe climb ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)




Simulating with [ hammer axe ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
```

```
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)




Simulating with [ hammer electric drill ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer electric drill rock ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer electric drill rock climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer electric drill rock climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
```

```
Simulating with [ hammer electric drill rock ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
```

```
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ hammer electric drill climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ hammer electric drill climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ hammer electric drill ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
```

```
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ hammer rock ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)
Simulating with [ hammer rock climb ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)
Simulating with [ hammer rock climb ladder ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)



Simulating with [ hammer rock ladder ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)



Simulating with [ hammer climb ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
```

```
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)
Simulating with [ hammer climb ladder ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)



Simulating with [ hammer ladder ]:
1 Zones. Min: 170, Max: 170
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 170
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 170
----brick vs hammer (min:5.14|mean:5.14|max:5.14)
```

```
Simulating with [ axe ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ axe electric drill ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ axe electric drill rock ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ axe electric drill rock climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ axe electric drill rock climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
```

```
Simulating with [ axe electric drill rock ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
```

```
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ axe electric drill climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ axe electric drill climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ axe electric drill ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
```

```
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)




Simulating with [ axe rock ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ axe rock climb ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ axe rock climb ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
```

```
Simulating with [ axe rock ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
```

```
Simulating with [ axe climb ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
```

```
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
Simulating with [ axe climb ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)



Simulating with [ axe ladder ]:
1 Zones. Min: 162, Max: 162
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 162
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 162
----brick vs axe (min:5.09|mean:5.09|max:5.09)
```

```
Simulating with [ electric drill ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ electric drill rock ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ electric drill rock climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ electric drill rock climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)



Simulating with [ electric drill rock ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)



Simulating with [ electric drill climb ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
```

```
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
Simulating with [ electric drill climb ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)



Simulating with [ electric drill ladder ]:
1 Zones. Min: 56, Max: 56
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 56
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 56
----brick vs electric drill (min:4.04|mean:4.04|max:4.04)
```

```
Simulating with [ rock ]:
1 Zones. Min: 51534, Max: 51534
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 51534
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 51534
----brick vs rock (min:10.85|mean:10.85|max:10.85)
Simulating with [ rock climb ]:
1 Zones. Min: 51534, Max: 51534
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 51534
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 51534
----brick vs rock (min:10.85|mean:10.85|max:10.85)
Simulating with [ rock climb ladder ]:
1 Zones. Min: 51534, Max: 51534
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 51534
--Attacker Deteced: N
--Visited: Y
```

```
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 51534
----brick vs rock (min:10.85|mean:10.85|max:10.85)




Simulating with [ rock ladder ]:
1 Zones. Min: 51534, Max: 51534
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 51534
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 51534
----brick vs rock (min:10.85|mean:10.85|max:10.85)




Simulating with [ climb ]:
1 Zones. Min: 32605775, Max: 32605775
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 32605775
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 32605775
----brick vs [No Valid Attack] -> Applying arbitrary high defense value
Simulating with [ climb ladder ]:
1 Zones. Min: 32605775, Max: 32605775
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
```

```
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 32605775
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 32605775
----brick vs [No Valid Attack] -> Applying arbitrary high defense value



Simulating with [ ladder ]:
1 Zones. Min: 32605775, Max: 32605775
Outer:
--Label: Outer
--Height: 20
--Width: 20
--Base: 0
--Final: 0
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
Zone[0]: Room 1
--Label: Room 1
--Height: 20
--Width: 20
--Base: -1
--Final: 32605775
--Attacker Deteced: N
--Visited: Y
--Prev: Outer
--[129320]Cerberus.Objects.WallClass Delay: 32605775
----brick vs [No Valid Attack] -> Applying arbitrary high defense value



==[Cerberus ending run @ 1/08/2013 4:05:50 PM]==
```