Risk analysis and communication for buildings using virtual reality

by

Vitalijs Terentjevs
BSc, Riga Technical University, 2013
MSc, Warsaw University of Technology, 2015

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Applied Science
in the Department of Civil Engineering

We acknowledge with respect the Lekwungen peoples on whose traditional territory the university stands and the Songhees, Esquimalt and WSÁNEĆ peoples whose historical relationships with the land continue to this day.

Supervisory Committee

Risk analysis and communication for buildings using virtual reality

by

Vitalijs Terentjevs
BSc, Riga Technical University, 2013
MSc, Warsaw University of Technology, 2015

Supervisory Committee

Assistant Professor David Bristow,
Department of Civil Engineering
Supervisor

Assistant Professor Ralph Evins,
Department of Civil Engineering
Departmental Member

Abstract

Traditionally risk management is associated with identification, evaluation and prioritization of risks. Nonetheless, communication of the risks to the parties involved is of the utmost importance. By providing more complete and easy to perceive information regarding potential hazard impacts and economic losses, risk analysis output increases risk awareness and helps make risk-informed decisions.

At present, in the field of civil engineering three-dimensional (3D) models are almost exclusively used for the design of structures. The presence of 3D and Virtual Reality (VR) technologies in risk analysis is extremely scarce. At the same time, there are potential advantages these technologies can provide to risk analysis and communication: the virtual 3D environment can emulate physical space and relationships between elements of the system, time-dependent simulations of hazard propagation, and awareness of physical dimensions of elements and their interconnection can be integrated.

This work is concerned with the way to communicate risks associated with building systems to decision-makers by visualizing them on a 3D model of construction and through simulation in a VR environment. For this purpose, the Almonte Power Plant in Mississippi Mills, Ontario, is analyzed as a case study. It is a small scale hydropower plant that is at risk of flooding, being located close to the Mississippi River. The last large scale flood in this region occurred in April 2019.

The novel methodology is applied to the aforementioned case study and further experiments are performed to test the sensitivity of the model to various parameters. The parameters of interest are flow rate and the degree of dependency between elements. Risk scores are obtained and evaluated as a function of flow rates and duration from the onset of flooding. The change in the degree of dependency between various elements of the electrical system allows an illustration of the importance of expert judgement of those dependencies.

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

Abbreviations

3D	Three-dimensional
4D	Four-dimensional (includes time)
AEC	Architecture, Engineering and Construction
API	Application Programming Interface
CAD	Computer Aided Design
CEO	Chief Executive Officer
HVAC	Heating, Ventilation and Air Conditioning
ID	Identifier
MC	Moisture content
PBR	Physically Based Rendering
PP	Power Plant
TRC	Typical Room Conditions
UI	User Interface
VR	Virtual Reality

1 INTRODUCTION

The resilience of buildings is an increasingly popular topic to many, be it academic researchers, building operators or occupants. The infrastructure of cities is changing rapidly and it inevitably affects the complexity of building systems. The concepts of energy efficiency, air quality, sewage treatment and occupant comfort are experiencing ever-growing attention of specialists in the field. As a result, the increase of interest in safe building operation, maintenance and the cost of accident recovery is also not surprising.

Risk analysis of modern systems is a complicated problem due to the interaction of many unknown factors. It involves the analysis of various interconnected systems that rely on each other. Failure of one of the systems can have far-reaching effects on a cascade of others. O’Neill (2013) provides a good example of how a bug in software resulted in the large power grid failure in Ontario in 2003. This single event indirectly resulted in consequences to the food, water, transportation and communication sectors.

Risk analysis is also complicated at the building-scale. This type of assessment aims to account for risks associated with things like hazards striking a single building or a small group of facilities. In recent years, the emergency management of facilities has become more important than ever before due to an increase in the density of cities and changing hazard possibilities. In many cases, empirical assessment is not viable and more sophisticated computer-aided assessment is appropriate (Rüppel and Schatz, 2011).

Effective decision making under risk relies heavily on risk communication with the involved parties. In general, by providing more complete information regarding potential disaster impacts and economic losses, risk communication can support analysis and decision making. This proves difficult in the design of buildings due to traditions embedded into planning processes in the architecture, engineering and construction (AEC) industry. These practices are characterized by a large number of participants and a high amount of planning information stored in distributed, heterogeneous and partial models of buildings. A typical list of systems of a building considered includes, but is not limited to, structural elements, heating, electric, water, communication, ventilation, and other assets within the premises. Each of those requires design by a specialist of a different field, which typically results in disconnected models and drawings.

The focus of this thesis is the potential flooding of a building, in this case a hydro power plant, and the effects of flooding on various building systems. It focuses on visualization and communication of those risks using state-of-the-art technologies. The goal is to investigate how novel technologies can be adapted to maximally increase the integration of risk information into a visual medium with the objective of increasing awareness of users to possible effects that floods can cause.

This is a two-part project. The first part develops a Virtual Reality (VR) simulation. The objectives of this portion can be summarized as being something that is:

1. Suitable for various flooding scenarios - the model has to be portable, not restricted to the example building studied herein.

2. Adaptability to different models – by (1) separating fluid physics and rendering processes to ensure that the physics can be refined in the future; (2) adjustable model complexity (i.e. inclusion of further hazards in the future); and (3) controllable (e.g., adjustable flow rate to support different hazard severities).

3. Responsive to geometrical restraints - the model is primarily intended for use in a building interior filled with various obstacles (e.g. machinery, furniture), the simulation has to respect geometrical boundaries and change its behaviour upon colliding with these solid objects. Water particle collision with solid objects ensures that the volume of the liquid is equivalent to the volume of the space. The law of conservation of mass is the foundation on which the simulation is built.

4. Able to interact with objects and systems. Since the model is used for risk analysis, objects must to exchange information with objects (upon collision or other forms of interaction). An object's reaction, which in this case is a risk of interruption or failure, depends on its properties.

5. Convey model outcomes via visual rendering - as mentioned before, the main goal of this is to convey risk information to a wide audience. As such, the three-dimensional (3D) rendering of flowing water is an intrinsic part of the simulation model.

The second part of the thesis is concerned with the application of the developed risk analysis model to the case study of the Almonte hydroelectric power plant located in the province of Ontario. The model is tested for several flood flow rates, as well as, for changing dependency between building elements in order to examine how the degree of coupling of building components might impacts risks. These tests allow for the evaluation of the sensitivity of the model to changes in important variables.

After the VR simulation and the risk analysis are described, a discussion of future work is presented. This is focused on the possibility of developing a rigorous user testing program. The testing should reveal the subjective evaluation of the software and provide future insight for development of future VR projects in the field of civil engineering and risk management.

1.1 Author's Contribution

This thesis describes a novel simulation and case study that will be submitted as a peer-reviewed manuscript. The author list, preliminary title and author contributions are:

Terentjevs, V., Atef, A., Bristow, D.. *Risk analysis and risk communication for buildings using virtual reality*.

- T.V., the author of this thesis and an MSc student at University of Victoria, developed (1) the workflow for conversion of a BIM model into a VR asset; (2) the VR simulation (i.e., the flood and risk components); (3) modification and use of a risk analysis client in Python; (4) the case study risk analysis; and (5) the manuscript.
- A.A., a post-doctoral fellow at University of Victoria, conducted and converted the laser scan of the building into a BIM model; provided the first draft of the power plant's risk model; and created the Python code for calling the risk analysis engine used in the risk analysis.

- D.B., an assistant professor at University of Victoria, supervised, provided overall direction and reviewed and revised writing

2 LITERATURE REVIEW

2.1 RISK COMMUNICATION

Traditionally risk management is associated with identification, assessment and prioritization of risks (Hu-Chen Liu et al., 2013; MITRE, 2014). An additional step of importance is communication of the risks to the parties involved (Eppler and Aeschimann, 2009). The goal of this activity is an improvement of effectiveness and transparency of risk management, as well as, encouragement of active participation of risk discussion (Brunner and Giroux, 2009). By providing more complete and easy to perceive information regarding potential hazard impacts and economic losses, risk analysis output increases risk awareness and helps make risk-informed decisions.

The visualization and communication of risks is an intrinsic part of risk analysis. Eppler and Aeschimann (2008) define risk visualization as: "the systematic effort of using images to augment the quality of risk communication along the entire risk management cycle". Presently, a variety of widely accepted graphical approaches are used to facilitate this task, among them: charts, risk matrices, network diagrams, spatial risk maps, traffic light systems, animated diagrams (e.g. GapMinder.org), and interactive visualizations (Eppler and Aeschimann, 2009; Roth, 2012).

A quickly growing area of risk communication uses 3D simulations in order to more realistically convey the risk information. Puschmann et al. (2016) noted that the accuracy of the simulation plays a significant role in risk perception by observers.

One of the challenges of providing a new form of risk communication concerns designing the experience. In the spirit of design thinking, the first thing to consider is the end-user. Each particular case may have a different target audience. The ultimate goal of this exercise is to make the product accessible to the appropriate audience by delivering how-to-guidance, visual information mediums and the ability to store and replay the experience.

In essence, engineering design is similar to a literary narrative. In a sense that it moves forward in time going through phases of a problem statement, solution development and production of outcome (Dusold, 2008). This process is well illustrated by Freytag's Pyramid

(Figure 1). In many ways, the state-of-the-art in risk communication in 3D representations of buildings experiencing hazards like floods, remains at the problem definition phase as is discussed in the following sections.

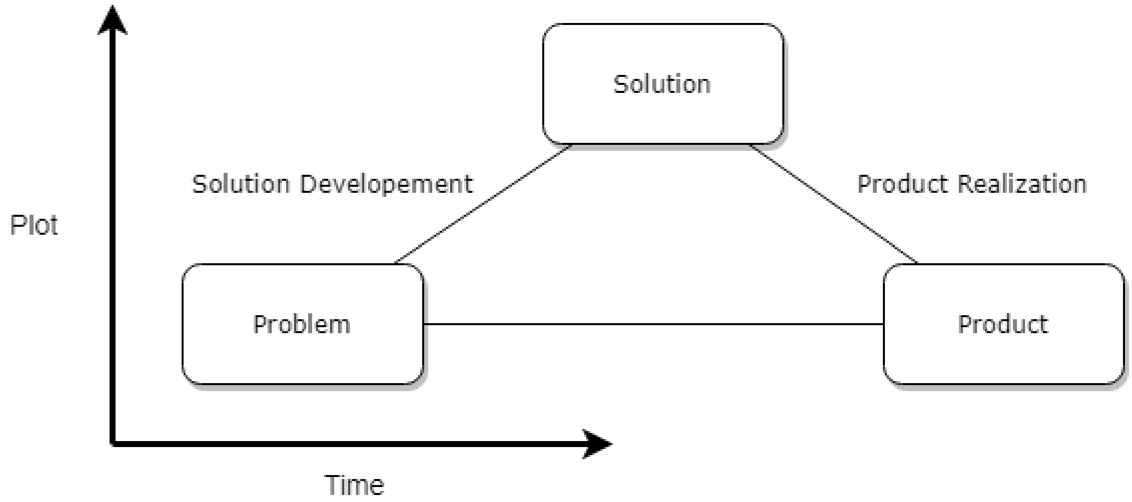

Figure 1: Engineering design process illustrated by Freytag's Pyramid.

2.1.1 Virtual reality in civil engineering

The use of Virtual Reality (VR) for civil engineering applications is still quite limited. Some of the engineering fields that greatly benefit from VR are architecture, construction, rehabilitation, maintenance, education, risk assessment, professional training, education and quality assurance (Puschmann et al., 2016; Sampaio et al., 2010). One of the leaders in VR technology development is Unity Technologies. In an interview with TechCrunch, Unity Chief Executive Officer (CEO) John Riccitiello says: “We’re increasingly supporting industries like architecture, engineering, construction and the auto industry” (Riccitiello, 2019).

Presently civil engineering design is predominantly done using 2D or 3D drawing software. Transferring geometry of a structure from the physical world into a virtual environment is associated with imprecision. VR applications can bridge this gap by providing an additional degree of realism to the virtual environment (Kumaran et al., 2007). As much as 3D visualization enables a better understanding of space compared to a drawing, incorporation of VR technologies makes the 3D model more tangible. (Sampaio and Henriques, 2008). Furthermore, the use of a real-time development platform adds a fourth dimension, time, enabling an analysis of the evolution of the model subject to stimulus.

Virtual Reality can be described as a medium meant to increase the quality of communication. It creates a sense of participation in the scene whereas classical 3D applications employ outside observation. In addition, VR encourages interaction with the abovementioned world (Sharma et al., 2018). Among the advantages of VR headsets is that the exact position of the user in the virtual world can be tracked by the system, unlike traditional design software (Kumaran et al., 2007).

Engineers, as a rule, are used to imagining their projects mentally and depicting them as 2D or 3D designs and schematics. The same cannot be said about outsiders, such as clients, who might experience difficulty reading numerical data, graphs and building layout plans (Hilfert and König, 2015). VR allows for an accurate immersive experience of human interaction with the surrounding environment (Ni et al., 2017). Among other things, VR allows the use of sophisticated visuals, kinesthetic and audio cues (Sharma et al., 2018). Experiencing the scene in Virtual reality positively affects cognitive performance, increases risk awareness and perception (Moreno and Mayer, 2002; Stanney et al., 2002).

Interoperability between Computer Aided Design (CAD) and VR systems is still complicated considering they were each designed for different purposes, the former for engineering use and the later for entertainment (Woksepp and Olofsson, 2008). Nonetheless, engineers that were exposed to VR models report the usefulness of such models (Rüppel and Schatz, 2011).

2.1.2 Virtual reality in Risk Analysis (communication)

In risk analysis, VR models are used to create mock-up scenarios of various hazards that allow users to experience risks and impacts firsthand. A VR model is, in essence, a depiction of a potential real-world setting with close-to-reality visuals and behaviour appealing to the user's imagination in order to produce an immersive experience (Puschmann et al., 2016). Fields interested in VR simulations are risk assessment, production planning, quality assurance, professional training.

Although the importance of virtual environments and immersion is still debated (Welch, 1999), various VR solutions are employed to train personnel in risk perception, potential hazard observation and identification. Exploration of hazardous scenarios can utilize both Virtual Reality and Augmented Reality technologies to enhance the experience. Additional benefits can be gained if haptic feedback is employed in tandem with VR (Ni et al., 2017). Another advantage of the VR approach is the possibility of tracking the user position and their interactions with surrounding objects.

It has to be stressed that VR models may have usability issues, such as disorientation, motion sickness, and limitations with the control devices (Sutcliffe and Deoul Kaur, 2000). Those drawbacks can be overcome by user testing and adjusting parameters of the model, e.g. environmental lighting, view angle, means of locomotion, etc. (Nickel et al., 2015; Reed, 2018).

Rising security concerns suggest that exploring potential hazards in a computer-simulated environment is strongly preferred to on-site exercises (Rüppel and Schatz, 2011). VR allows access to areas otherwise inaccessible or dangerous to explore in real life. Potentially life-threatening simulations (e.g. earthquakes, fires) can be performed without endangering the user (Kumaran et al., 2007). For researchers, it means conducting hazard related trials with human subjects in compliance with national and international ethics committee requirements.

2.2 Building floods

ANSI/IICRC S500 (IICRC, 2015a) defines three categories of floodwater: sanitary water, water with significant contamination, and grossly contaminated water. In reality, category 1 water rapidly deteriorates to class 2 or 3. This occurs because parts of structures have either been contaminated before the flood or the exposure time is long enough to cause contamination, such as mould. In the scope of this project, we examine category 2 water flooding as it is the most realistic scenario. Category 2 water is characterized by a potential danger to human health if ingested or contacted. It can contain inorganic pollutants (e.g. washing machine or dishwasher discharge) or organic pollutants (e.g. toilet bowl overflow, broken aquariums).

Further, ANSI/IICRC S500 defines 4 water intrusion classes based on the type of the materials used in the floor, wall and ceiling construction: (1) less than 5% of materials are porous (e.g. carpet, gypsum board, textiles, porous concrete, mineral wool, etc.), (2) 5-to-40% of materials are porous, (3) greater than 40% materials are porous, and (4) cases where water penetrates into low absorption materials (e.g. timber, concrete, masonry, etc.).

It is assumed that in many cases full restoration of structural elements is possible by replacing damaged portions of materials. According to Shaughnessy et al. (1999), microbial growth should not be expected in materials if water activity, i.e. relative humidity, and is related to the moisture content in materials (

Figure 2), is below 0.65. For assurance, the limit is considered to be 0.60 (IICRC, 2015a).

Heating, Ventilation and Air Conditioning (HVAC) systems may contain elements that cannot be restored cost-efficiently and have to be replaced after water penetration, especially if the water is contaminated. These elements may be a part of insulation, electrical or mechanical systems.

Personal property or contents include, but are not limited to, furniture, clothes, paper goods, appliances. Many of those are not initially damaged by the flood but their condition is dependent on exposure time.

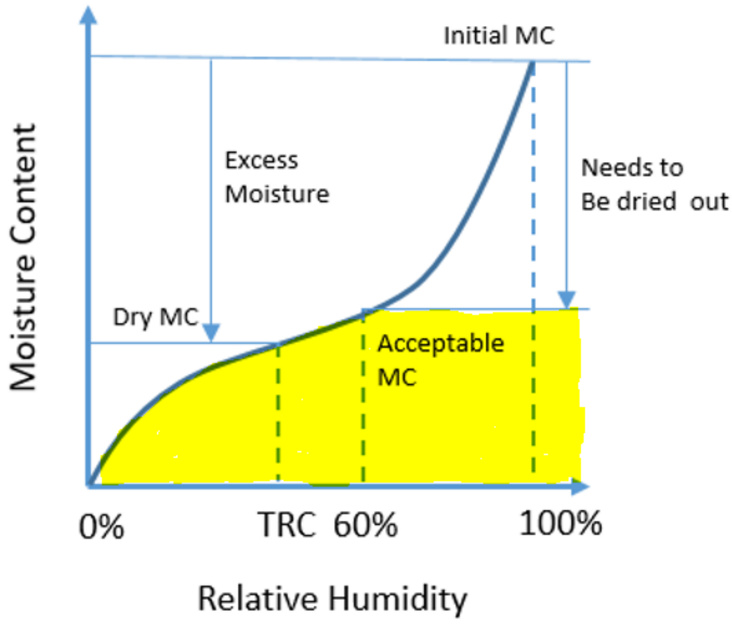

Figure 2: Qualitative representation of the excess moisture in materials (TRC – Typical Room Conditions, MC – moisture content.

Water by itself can have an extremely negative effect on electrical equipment. Flood water, in addition, contains salts, metals, acids and pieces of debris that may cause even more long-lasting damage. Electrical equipment can be categorized in the following way: conductors, wires, insulators and mechanisms. Switchgear, for example, may contain many of the aforementioned. Particularly copper control wiring that can continue to react with acids for a long period of time leading to corrosion. Places, where conductors meet insulators, are endangered by water entering insulation. Floodwaters are a tremendous danger to mechanisms because they typically consist of many moving parts, such as bearings, pins, rings and latches. Malfunction of rotating mechanisms can be an immense danger to human safety. As a result, even if the mechanism was not directly submerged in water it may be damaged indirectly and be a risk factor. If contaminated water has entered the mechanism directly its complete disassembly is necessary and considering a complete replacement may be needed (Genutis, 2013).

2.3 WATER DAMAGE

Though there are several methods of evaluation of water damage to a property, as described in various standards (IICRC, 2015a) (IICRC, 2015b). Estimation of potential damage is complicated by the need to consider factors such as stream velocity, water contamination, sediment transport, duration, and response (Pistrika et al., 2014). Zhai et al. (2005) have developed a model that considers water depth, house ownership, type of a house and household income in water damage evaluation. Thieken et al. (2005) considered resistance of elements to water damage and flood depth as two important factors. A study by Wind et al. (1999) investigated the influence of risk response time. Merz et al. (2013) suggested parameterless machine learning algorithms to assess the damage. In practice, damage estimation is based on depth-damage functions that relate water depth to the economic value of the damage. USACE (1982) defines the damage percentage for evaluation of flood damage as follows in equation (1). In the equation, market value is an outcome of professional evaluation of the building, and the cost of repair is the expenses necessary to restore this value. The damage percentage is a numerical value between 0 and 1.

$$Damage\ Rate = \frac{Cost\ of\ Repair}{Market\ Value} \tag{1}$$

where Cost of Repairs and Market Value of Property should be marked at the same time period considering a constant change of retail value.

Xinyu Jiang et al. (2013) expand the previous equation to obtain the risk value in equation (2).

$$Risk = Property \times Damage\ Rate \times Probability\ of\ Water\ Depth \tag{2}$$

In the case the Property term (i.e., the Market Value of the Property) value is unknown and the Probability of Water Depth is 1, the equation can be reduced to equation (3)

$$Risk = Damage\ Rate \tag{3}$$

The damage rate is obtained either as a result of regression analysis of statistical data or evaluation of expert opinion. Typically only the independent variable in the analysis is water depth. Among other risk assessment manuals, the United States Army Corps of Engineers (USACE, 2015, 2006, 2003) provides guidelines for damage rate evaluation of various structures.

Damage functions for the North America region as seen in the JRC technical report (Huizinga et al., 2017) can be found in Table 1. The values in the table correspond to the level of damage where 0.00 stands for no damage and 1.00 for complete loss of property. Values for residential commercial and industrial buildings are given. In addition, Figure 3 depicts the damage curve comparison between North America and Europe.

Table 1: Damage functions for North America

	Damage factor		
Floodwater depth, m	Residential buildings	Commercial buildings	Industrial buildings
0	0.20	0.02	0.03
0.5	0.44	0.24	0.32
1	0.58	0.37	0.51
1.5	0.68	0.47	0.64
2	0.78	0.55	0.74
3	0.85	0.69	0.86
4	0.92	0.82	0.94
5	0.96	0.91	0.98
6	1.00	1.00	1.00

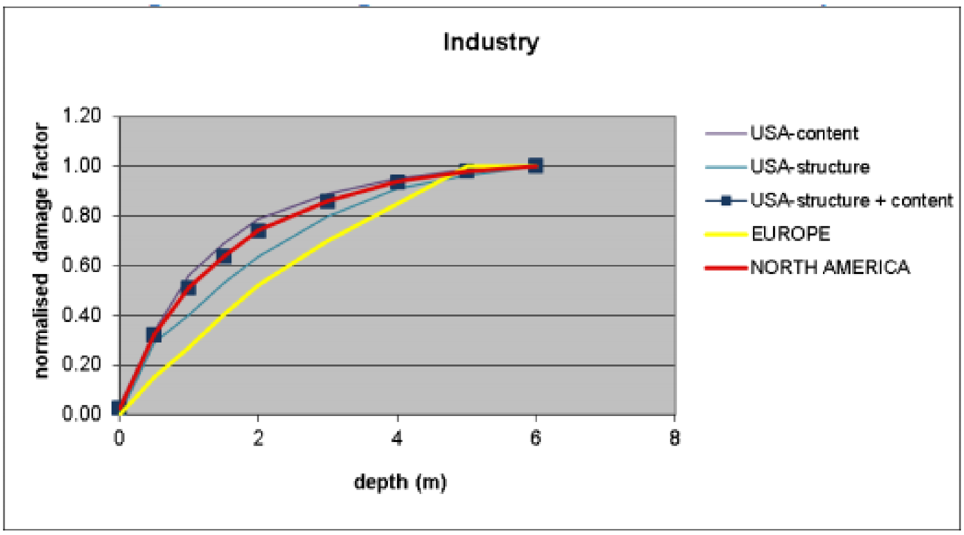

Figure 3: Damage factor comparison for industrial buildings (Huizinga et al., 2017).

2.4 BUILDING INFORMATION MODELS

Increasingly, modern buildings are designed with the help of BIM. In the present state, BIM is primarily a design tool, nonetheless, its incorporation into the engineering process provides long term benefits for building maintenance, retrofitting and modification. In the scope of our project, it also contains vital data for risk analysis.

The main drawback of BIM is that it is still emerging technology and most of the existing structures do not have complete models. Technical documentation for such instances may be available in the form of CAD drawings or paper blueprints. Considering there is no ready-available software to convert such documents into BIM the process is time-consuming and susceptible to mistakes. Another solution is the point cloud scanning of facilities as-built. This allows the creation of more true-to-life models but is limited only to the objects and systems on an element's surface.

Among the data integrated into BIMs, several aspects of it are of interest for risk analysis and 3D rendering. First is the physical dimensions and position of elements in relation to other objects. Second is the materials used in and the properties of those materials. Properties of interest might be mechanical strength, fire-resistance, water-tightness, chemical composition, etc. From a data management point of view, essential information provided by BIM also includes element names and unique identifiers (ID). These latter two pieces of information are useful in many ways, such as for asset identification in models, databases and outputs.

One of the examples of BIM software is Autodesk Revit. It was chosen for the case study in the framework of this thesis as one of the most popular building information modelling engines. In later chapters, the data extracted from Revit and its processing are discussed in detail.

2.5 DEPENDENCY MODELS

Visualizing and communicating risks of buildings first requires an assessment of risks. Engineering is comprehensive in methodologies for assessing risks to individual components or to individual systems of buildings (e.g., structural analysis). An increasing area of risk relevant to modern buildings with interconnected systems is the risk of failures due to the dependencies between components. The failure of an element can result in a cascade of events and subsequent failure of other elements of the system. This sequence can result in the inability of a building to fulfil its functions. Tracing these cascade effects can be problematic. Atef and Bristow (2019) provide a methodology to extract the necessary data from BIM and use it in the development of models useful for assessing risk among the interdependencies of components.

The connections between different elements of a system or several systems may not be obvious and engineers of different backgrounds might give more emphasis to different aspects (e.g. architecture, HVAC, safety systems). Atef and Bristow (2019) emphasize two types of interdependency models: spatial and operational.

The spatial dependency model is based on a location of elements in a space of a building. The building is divided into rooms and all the objects that are physically located inside the room are a part of the system. Each element can be part of several groups, such elements are located on boundaries of spaces (e.g. walls, doors). In addition, adjacent rooms have to be established to make a spatial model. Several of the commercially available BIM packages have Application Programming Interfaces (API) to determine object position in relation to reference points, which makes this type of data extraction possible.

The operation dependency model is best suited to describe HVAC systems, where each system has a source, machines or appliances, and connecting elements. Those elements create a network and are connected to each other in a specific order. In many cases, the direction of dependency is from the preceding-to-succeeding element, but bi-directional connection is also possible. This latter case can be exemplified by a diesel generator where a failure of either generator or control panel results in the disruption of the other, while the failure of the electric outlet does not affect the control panel (Figure 4). Some systems might include elements of different purposes; such as an electrical network connected to the boilers supplying heat. The important systems are expected to have reserve sources.

Figure 4: Simplified scheme of Diesel generator.

The risk and dependency model is based on the strongest path method described by O'Neil (O'Neil, 2017, 2013). The method employs three variables: (1) likelihood of occurrence, (2) degree of impact and (3) degree of dependence. The values of those variables are inferred from data and expert judgement. To estimate the impact of every entity on every other entity of the dependency graph, path analysis is used. The path analysis itself is an extension of multiple regression. Its purpose being provision of significance and magnitude to causal connections (Webley and Lea, 1997).

O'Neil's method is available in software called RiskLogik. Upon completing the analysis there are several ways to interpret and visualize the outcome. Methods available in RiskLogik for understanding the risk are Path analysis, Risk Plot, and Risk Stack.

Path analysis allows exploring dependency paths between various elements of the network based on criteria, such as length, strength and degree of cumulative dependency. This particular diagram presents a logical chain connecting two nodes of interest through the sum of intermediates.

The risk plot produces a graph of the cumulative likelihood of failure plotted against cumulative global impact. It provides statistics of each individual element of the system comparing to its counterparts. The risk stack essentially illustrates the risk index in equation (4).

$$Risk\ \ Index = Cumulative\,Likelihood\ of\ \ Failure \times Cumulative\,Impact \tag{4}$$

3 METHODOLOGY

Presenting the results of a risk analysis is closely tied to the problem of the creation of a narrative of the project. Outside of the final result, engineers are also interested in the process of achieving the target. The idea is that the observer is actively participating in the data presentation. For this purpose, interaction with data is highly desirable. An ability to create your own story by selecting input parameters (e.g. flow rate) allows the participant, whether the client or engineer, to understand their data in depth (Grant, 2018). Hence, the problem-solution mechanism serves the purpose of risk communication as well. Especially considering that modern-day projects are a collaboration of several specialists, frequently of different backgrounds and our project is not an exception.

Using the pattern described in chapter 2.1, the project is developed in iterations where each outcome raises a new problem to improve upon. This design approach allows for revisiting the project at any stage of development and monitoring design evolution. This provides a good platform for analysis of advantages presented by different solutions used.

Applying theory to our project we were looking for a work-flow that connects different pieces of software (including those developed herein). Following this process, individually tailored solutions for particular hazard scenarios can be created. The diagram of the risk modelling process is presented in Figure 5.

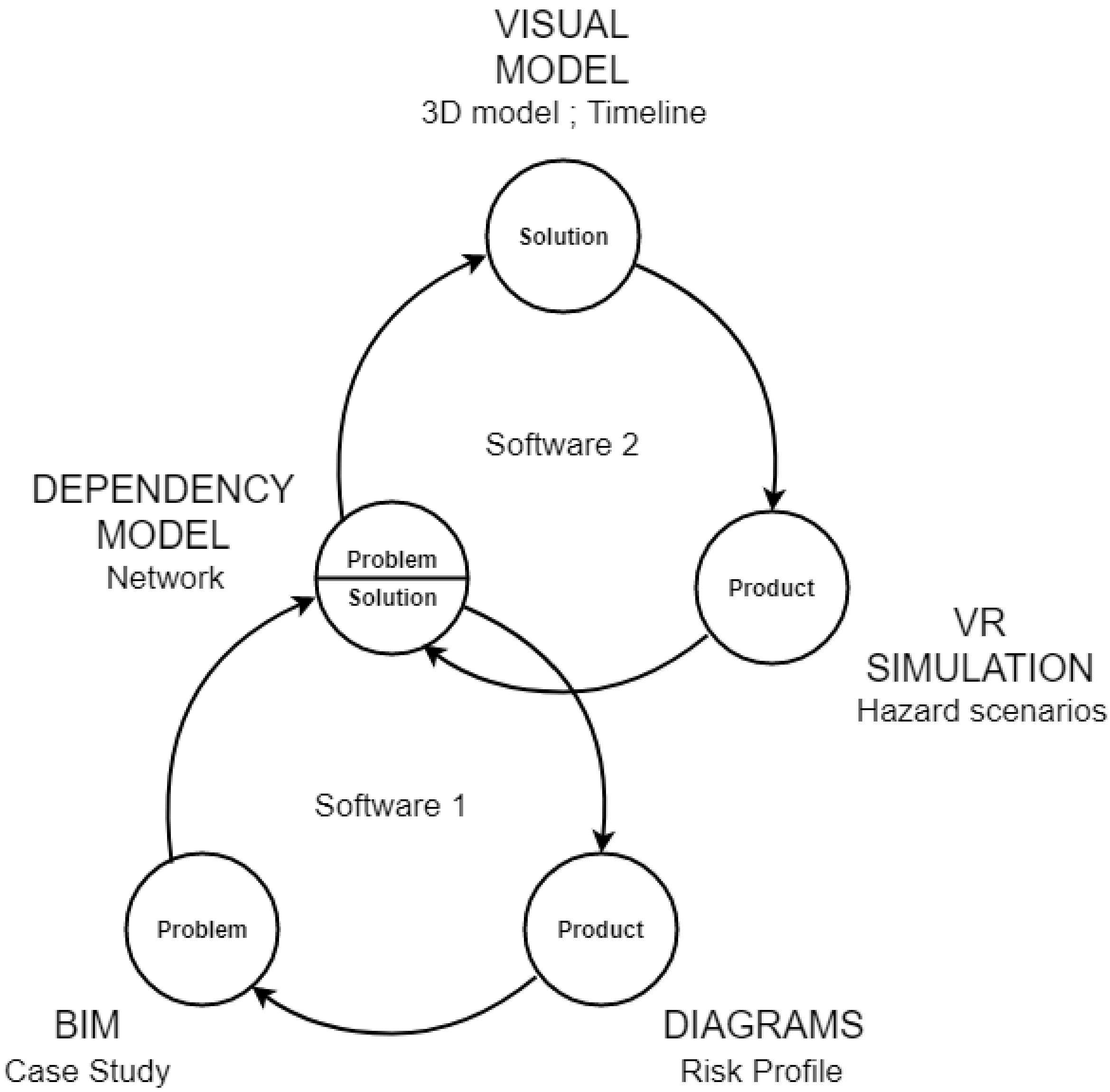

Figure 5: VR risk simulation model's design flow.

Each particular case is developed by following the flowchart. The process is essentially divided into two principal pieces: risk model development module (1st module), and risk visualization module (2nd module). The development procedure of each module follows the aforementioned Freytag's Pyramid's pattern consisting of three principal parts: problem, solution and product (output). Noteworthy is the fact that the solution to the risk module presents a problem for the visualization module.

The cornerstone of the risk model is the Building Information Model. This is the entry point of the first module where most information about the design is input. This is followed by the development of the dependency model, which is the basis for risk analysis and results in analysis output. Depending on the risk model outcome, changes can be made in the initial problem statement. BIM is essentially the cycle point where the geometry of a building and location of assets can be adjusted.

After the dependency model is created and an intermediate solution to the risk model is found it is supplied to the visualization module. At this stage, a 4D (three dimensions of space and one of time) simulation is developed to visualize numerical data acquired from the first module. Building on this model another outcome is produced presented in a form of VR experience. Observing these results, all involved parties can express their expert opinion and make adjustments to the risk model. In turn, it leads either to modification of the dependency model or returns to the first cycle to make changes in BIM. Performing these iterations over two modules both products are revised.

In the following sections, the specifics of each process cell are discussed in detail. In particular this includes, the workflow of Software one: on the preprocessing of the risk model. The software for this consists of BIM created in Autodesk Revit, RiskLogik – a risk model engine producing a comma-separated value output interpreted using Microsoft Excel. Further, the process of integration of the risk model into the Unity Engine (Software 2) is described and visual output in the form of VR is presented.

3.1 VISUAL MODEL

The visual model is an attempt to interpret numerical data in an easy-to-grasp medium. The purpose of it is to give space and time context to previously developed risk models in a way that is intuitive. The visual model is able to give natural answers to questions, such as:

- What is the distance between objects?
- Do objects have clear line-of-sight between them or is the view obstructed by barriers?
- What are the physical dimensions of objects?
- What is the object's relative position to the floor level?

In addition, the model is designed to present various risk analysis variables, i.e. cumulative risk score, degree of dependency between elements.

One of the strong points of using a real-time graphic engine is the ability to convert risk models from discrete time steps into an apparently continuous timeline within the simulation (Figure 2). This ensures a smooth transition between model states and an accurate representation of hazards. Aa a result, the simulation can be performed in a fluid motion from time point zero to the end or otherwise explored at any particular time point.

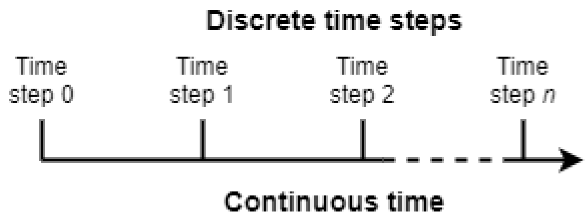

Figure 6: Continuous the timeline used in the visual model.

The visual stimulation is created by combining BIM and risk analysis data. The model creation pipeline is illustrated in Figure 7. The process of converting BIM into a conventional 3D format consists of several steps: (1) getting graphical data and refining it in specialized software (e.g., 3DS Max); (2) preparation of real-time engine specific data for visualization, such as materials and lighting; and (3) implementing features necessary for simulation, i.e. colliders, VR controls, hazard simulation components. The BIM data is uploaded to the database. Using this information, a risk model is developed and merged with the 3D model once more.

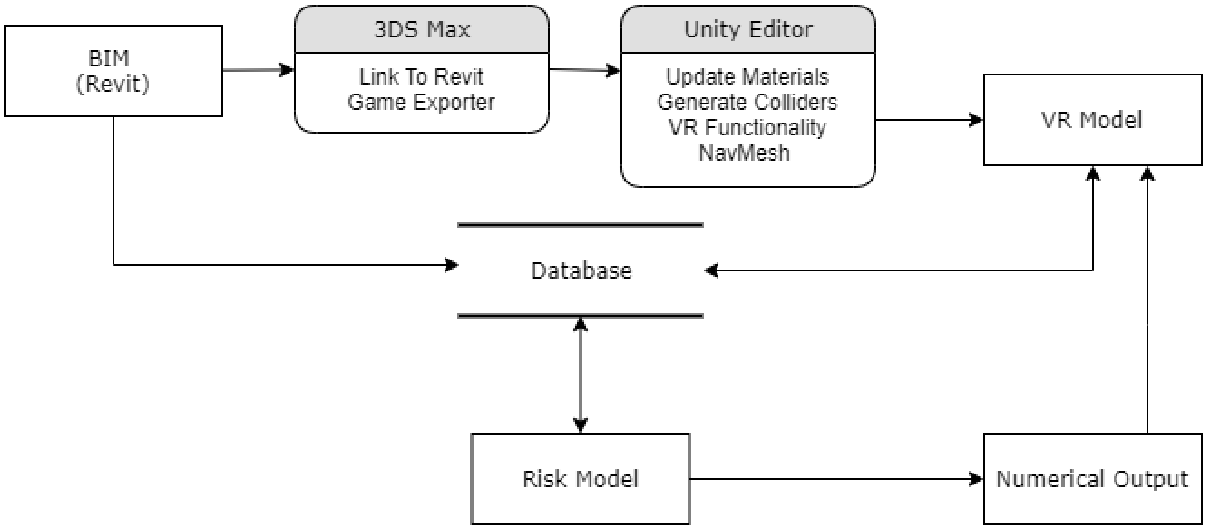

Figure 7: Visual model creation pipeline.

An important factor in this application is its flexibility. To achieve it the methods employed have to be universal and appropriate to different hazard scenarios and building layouts. For this purpose, coordinating communication with the database is necessary to ensure that most of the visual data is generated at runtime with minimum input from the user. As a consequence, a number of model elements are pre-processed. Among such entities are database files containing information about elements and relationships between them. The elements and connections are directly parsed from Revit using a specifically created plugin (Atef and Bristow, 2019). The other predesigned element is a risk model output prepared using RiskLogik software. The Jupyter notebook showing the use of the model is available in Appendix A.

Choosing the software, i.e. Autodesk Revit, Autodesk 3DS Max and Unity, several criteria were considered, such as (1) popularity among stakeholders; (2) interoperability (i.e. Revit and 3DS Max are products of the same company that is a long-time partner of Unity Technologies); (3) cost of licenses.

3.2 VR Simulation

Design thinking demands that any project development involves a series of staples, such as a user-centred approach, early prototyping, outside testing, small-to-big thinking, and cyclic design (Brown, 2008). Creating a VR model should firstly ask: "Who is going to explore the model?". By answering this, the level of complexity of the model, as well as the necessary visual presentation is determined. The development of the model starts with a simple abstract risk simulation and grows into specific hazard visualization scenarios, in our case flood simulation. The VR experience is then refined by testing and iterating on the user feedback. In the case of this project, the feedback from our industry partners.

Currently, the project is 25 novel and specifically developed classes and several open-source resources to help in the development of VR. An example of VoxelGenerator.sc script is included in Appendix B. The code is targeting the .NET Framework 4 developed by Microsoft and is employing Unity as a visual rendering engine.

3.2.1 Risk visualization

To achieve the desired effect of easy-to-grasp risk communication via VR the following components have to be created: (1) an intuitive 3D User Interface (UI); (2) time controls for the simulation; (3) visual deterioration of elements of the systems the depicts the cause of the deterioration; (4) statistical assessments; and (4) reusability.

Our idea to communicate risks associated with building systems is by visualizing them on a 3D model of a construction. The accuracy of the simulation plays a significant role in risk perception by observers. Firstly, this is accomplished using a 3D object whose geometry is easily recognizable (e.g. furniture, heavy machinery). Secondly, the materials used in the rendering are accurate depictions of their real counterparts. Paired up with VR navigation throughout the premises gives a feel of immersion in the scene.

To illustrate the probability of failure of elements, colour-codes are used. The workflow of the code assignment is presented in Figure 8. The hazard scenario stored in the database is loaded and cross-checked with an object in the simulation. Risk score analysis is performed in discrete time. At each timestep, a risk score is evaluated and updated accordingly.

One of the four colour codes is assigned: green (0-30%), yellow (30-50%), orange (50-70%) and red (70-100%). The custom-made rendering shader is created specifically for this project. The Unity Engine uses Physically Based Rendering (PBR), a technology pioneered by Disney and Pixar studios. This workflow has a number of channels representative of the physical properties of materials related to interaction with light (Figure 8, bottom). One such channel is "Emission"; it can be accessed procedurally and make an object glow in the specific colour in accordance with our colour-code.

Information delivery is additionally enhanced by the "world space" UI (i.e. UI positioned in a virtual world among 3D objects) intuitively paired with each individual element (Figure 9). The UI is set up to show the object name to make cross-references with other media formats (e.g. diagrams), risk score and impact score.

The strength of dependency between elements is shown by straight lines and assigned values between 1 and 10 (Figure 10). The glowing spheres are moving in the way that the thicker end is pointing towards the entity that is dependent on the entity at the thinner end.

To create the visuals for dependency lines the following process is employed (Figure 11). When raycaster (i.e ray-surface intersection testing mechanism, where ray is a mathematical representation of world space line, which is perpendicular to the camera view and intersects the mouse click point) returns a collider hit confirmation the algorithm tries to determine an ID of an object associated with the surface (the same ID used by Revit software to identify objects in BIM). Then the script calls the database to retrieve the list of connections for the hazard scenario visualized at the moment and tries to match it to the ID of the hit object as described in the "Get Con-ons" operator. A connection is an array of three values: IDs of elements on both sides of the connection and dependency strength. An object can be part of any number of connections. At the next step, the dictionary (dictionary of all building-type objects in the scene is created at the start of a simulation) is checked to confirm the existence of the objects in the scene. If both objects exist, their centre position in 3D space is used to create a line object in-between. According to the degree of dependency, the colour code is assigned to the line as described in Figure 10, where 1 corresponds to low degree of dependency and 10 to high.

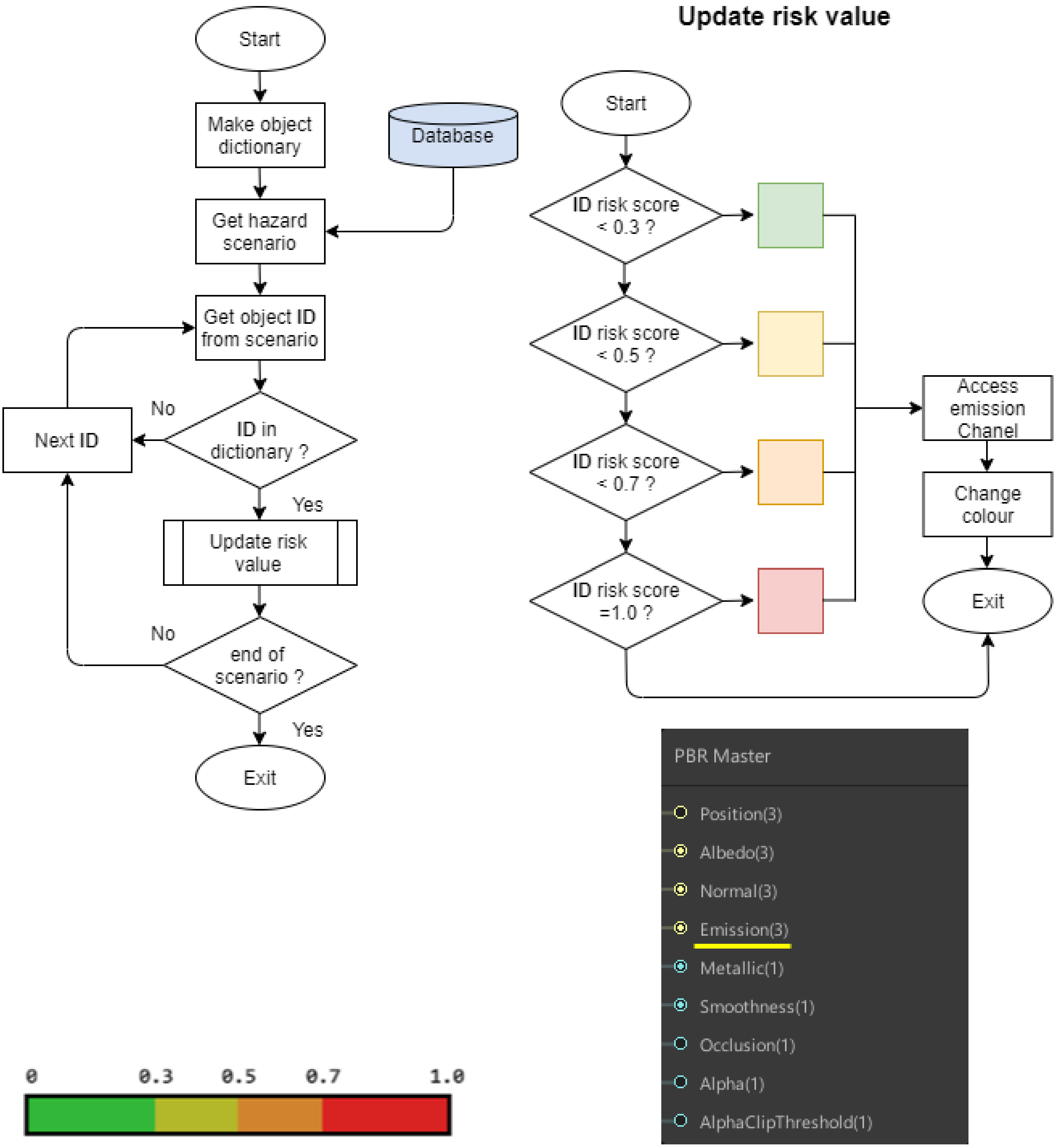

Figure 8: Colour-coding workflow.

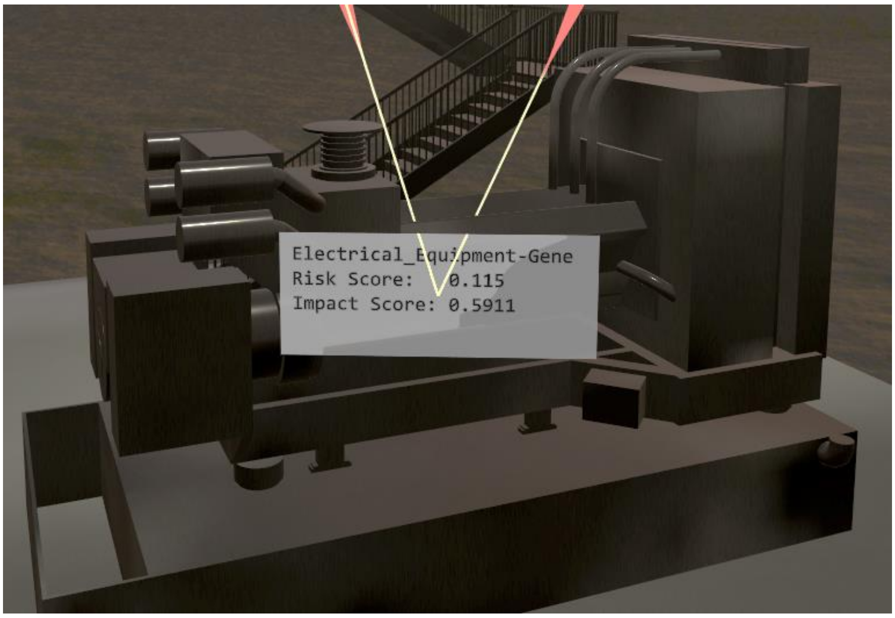

Figure 9: World UI showing information about Electrical Equipment –Turbine.

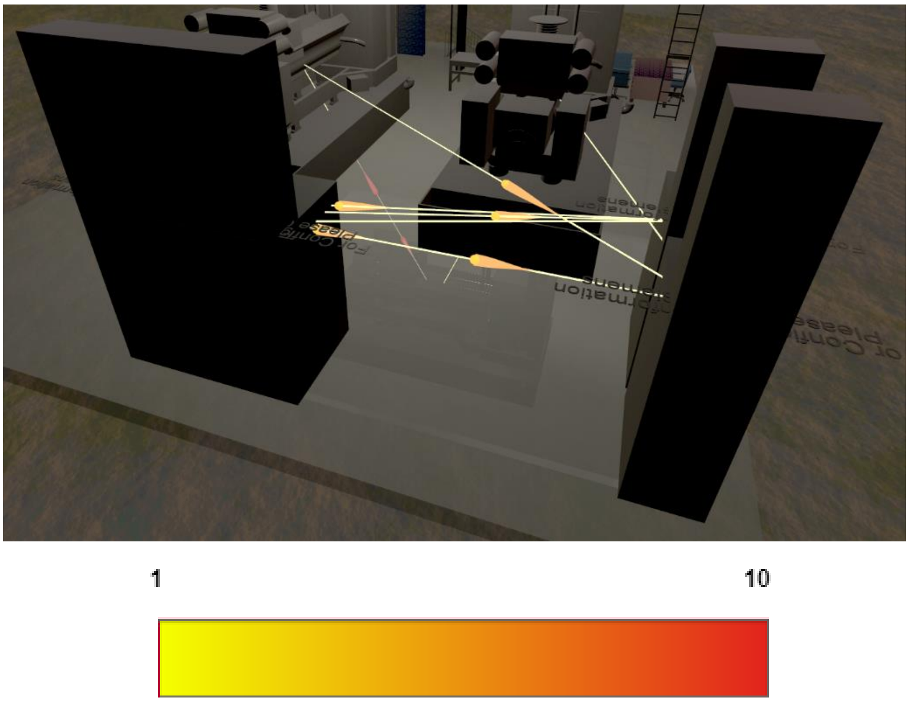

Figure 10: Visualisation of dependency in 3D.

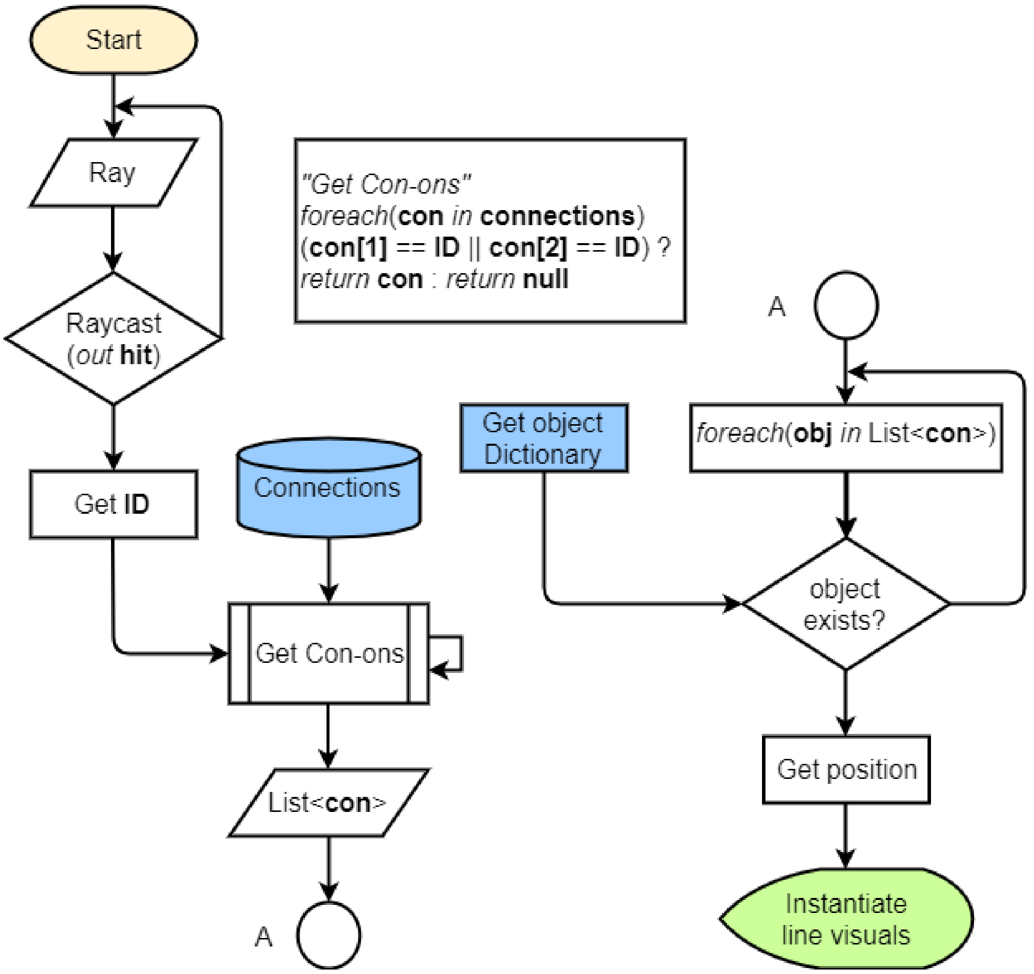

Figure 11: Flowchart of visualisation of dependency lines.

3.2.2 Flood simulation

Fluid simulation in computer graphics is primarily realized either through particle-based simulation or grid-based simulation (Braley and Sandu, 2010). For the purpose of this project, grid-based simulation is a better fit. Since this simplified water-flood model does not need to capture all of the fluid dynamics. Instead, emphasis of the model is given to parameters such as flow rate, spill area, water depth, and water volume. Monitoring all of them allows performing a volume-controlled simulation.

One of the key features necessary for accurate flood simulation is establishing water volume geometric boundaries. This is achieved through a collision detection mechanism provided by the real-time simulation engine. The process of capturing the geometry of a closed facility is illustrated in Figure 12 and Figure 13. It is based on the voxel matrix (named following an analogy with two-dimensional pixels in a bitmap image, a voxel is a cell in three dimensional space), which is iteratively updated using a cellular-automaton algorithm as follows. Initially, the matrix seed is planted in 3D space in a position relative to a horizontal surface (e.g. the floor) where water can flow unobstructed. It is considered to be the centre of the matrix and its special coordinates are recorded. After the cellular-automaton is launched it spreads outwards from the centre in horizontal water plane evenly in every direction. Upon encountering any obstacle (e.g. wall) the collision is detected and propagation of water in the direction of collision is terminated by assigning a negative value to the matrix cell. This mechanism allows accurate determination of the boundaries of the facility and consequentially the water spill area. Further, water propagation in a vertical direction is based on equation (5).

$$Depth = \frac{Area}{Flow\ Rate \times Time} \quad (5)$$

where Flow Rate [m^3/s] is determined on a case-by-case basis, Time [seconds] is calculated since the beginning of the simulation, and Area [m^2] is water spill area determined using the cellular-automaton algorithm. Depth [m] is directly correlated to Damage Rate (equation 3) through depth-damage functions. It has to be noted that this model does not include the possibility of water absorption by porous substances, nor displacement of buoyant objects by water, which can potentially make alterations to equation (5). Nonetheless, the author considers those to be insignificant in comparison to the test water volumes in this thesis.

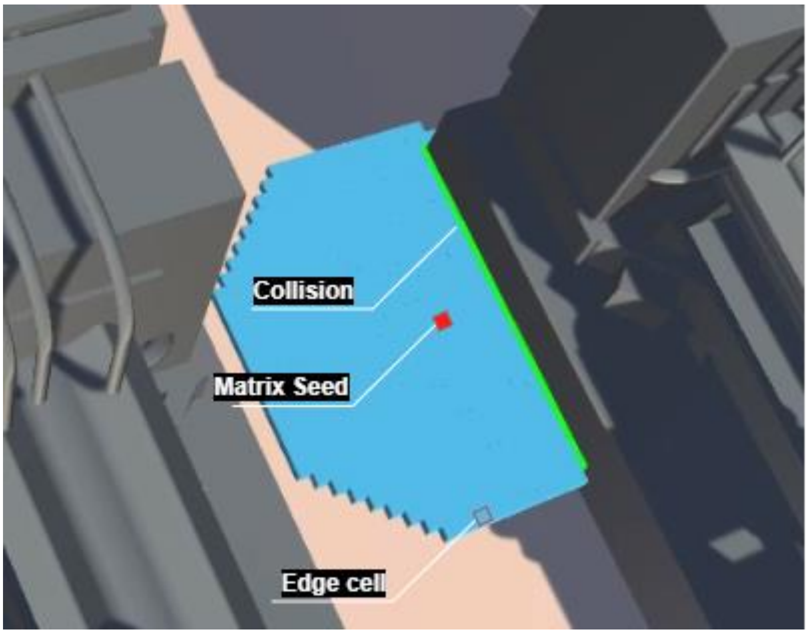

Figure 12: Cellular Automaton algorithm determines the shape of water spill.

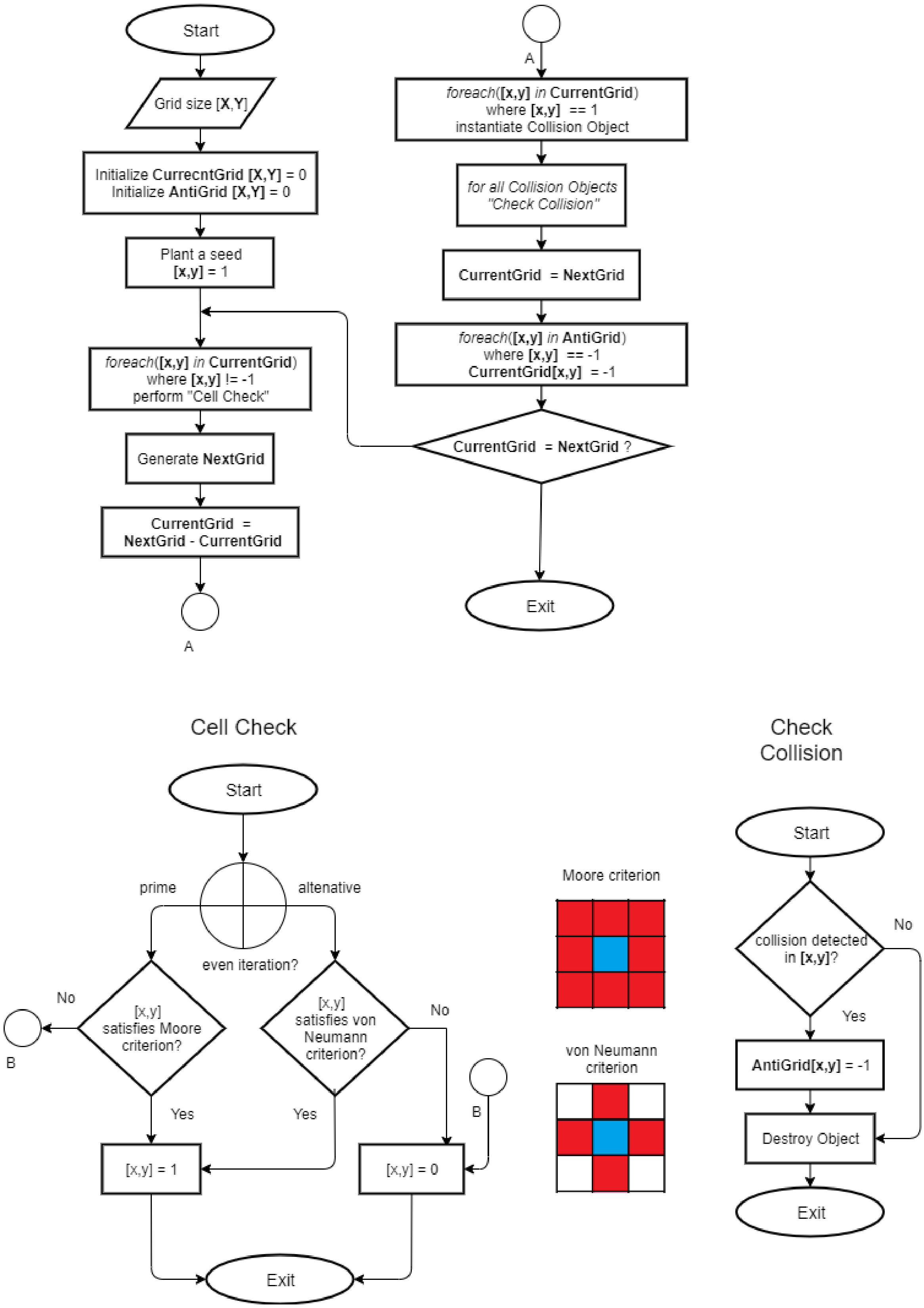

Figure 13: Water propagation algorithm

4 CASE STUDY RISK ANALYSIS: ALMONTE POWER PLANT

4.1 INTRODUCTION

The study follows from two objectives: (1) improvement of risk analysis and (1) easy-to-interpret presentation of the study outcome. The general solution is applied to the Mississippi River Power Plant in Almonte, Ontario. The essential function of the facility is power production. As a consequence, electrical systems and machinery are of the utmost importance to the operational process of Power Plant (PP).

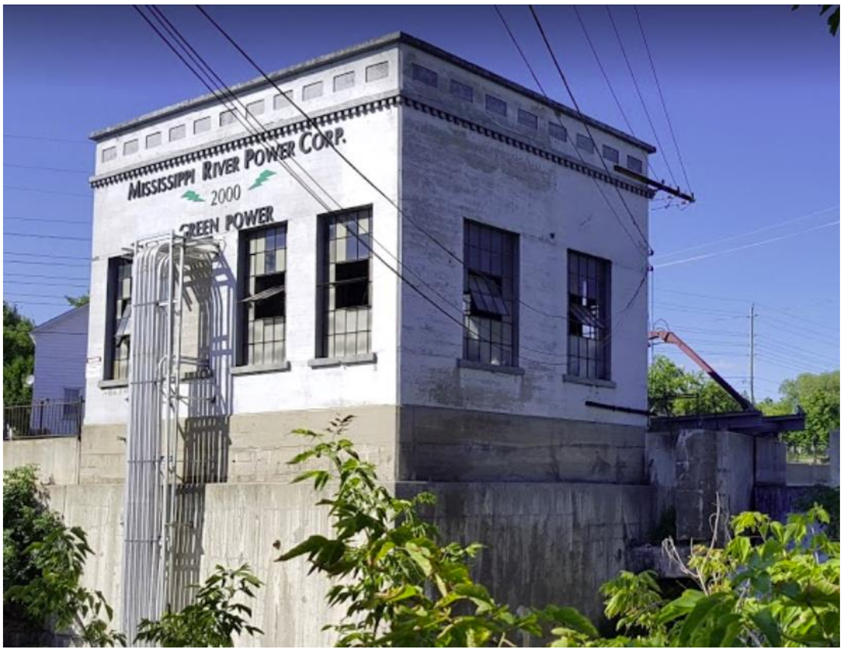

Figure 14: Almonte Power Plant, Almonte, ON. View from the outside (copyright: Google Maps/Google Earth).

The PP is located in Mississippi Mills, a region known for frequent floods. At the time of writing, the most recent flood occurred in April 2019 (Gesner, 2019) with as many as 6,500 homes flooded by April 29 (FLOODLIST NEWS, 2019). This flood did not directly affect the Almonte PP interior, though future flooding may. There is no information available about facility waterproofing to ground or overland waters, as a result, we assume that both are possible.

4.2 BUILDING INFORMATION MODEL

The BIM of the Almonte power plant was created using 3D laser scanning (the technology to capture and record the shape of physical objects) of the interior of an existing building. The result of this activity is a point cloud (Figure 15, a set of data points in 3D space representing the geometry of facility) that can be broken down into separate physical objects and interpreted by an expert. The following step is surface reconstruction and separation of model into distinct elements.

The 3D model of the plant is shown in Figure 16. Plans of the building with furniture can be seen in Figure 17 and Figure 18. Ground floor includes generator room, office, bathroom and entrance hall. Basement includes only turbine room.

Summary of the elements in project is presented in Table 2. The first column of the table contains element name as it appears in Revit (in Revit name consists of family name plus type); the second column is element category, which allows determination of object operational groups for future analysis; and the third depict the element position in space of the building necessary for spatial dependency establishment.

To ensure proper rendering of objects information about materials they are made of is retrieved from BIM. Material degradation is out of scope of this project, as such simplified material data is retrieved purely for the purpose of visualization. List of materials consists of concrete, aluminum, steel, brick, textile, porcelain, wood and plastic.

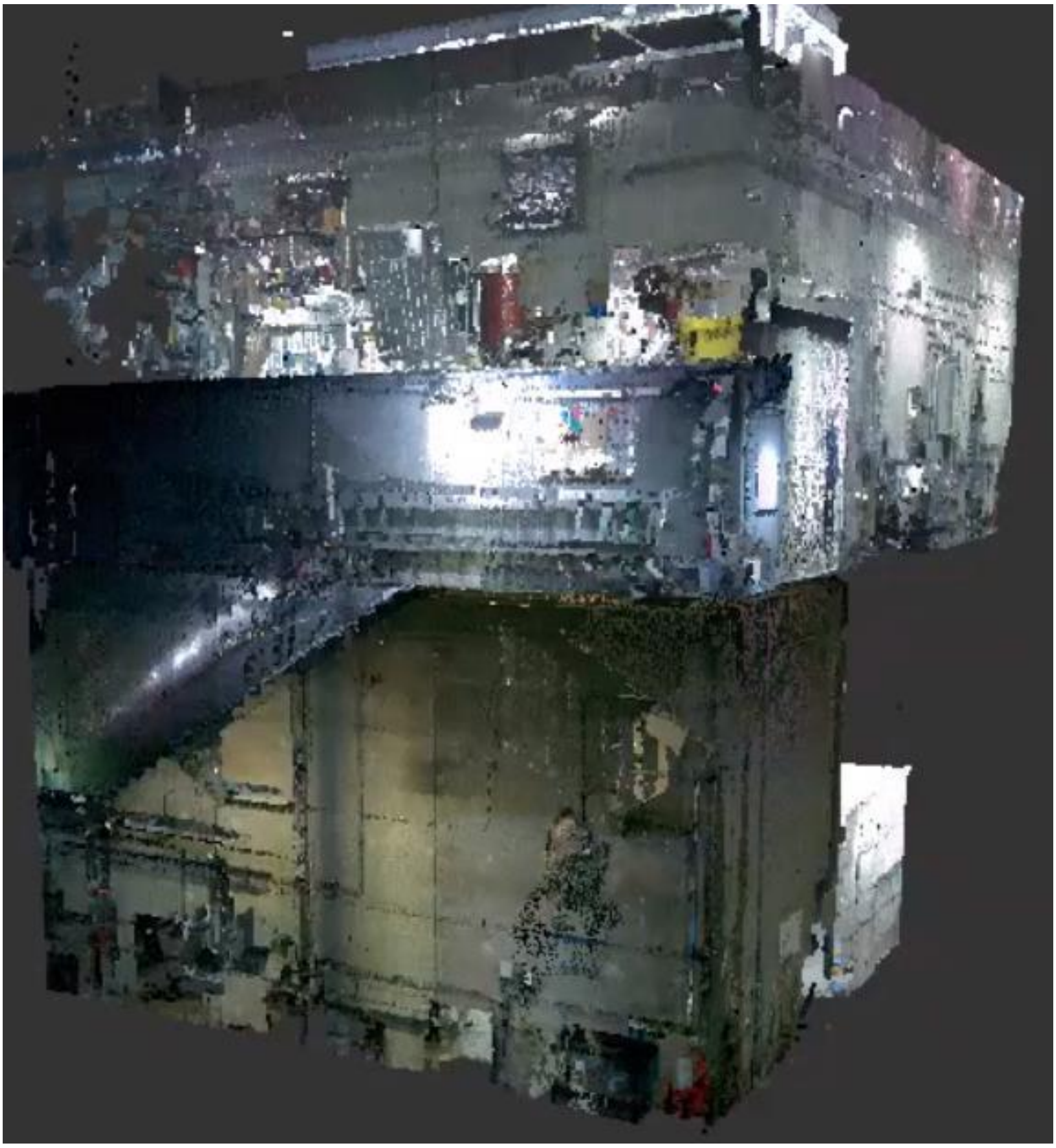

Figure 15: Point cloud of Almonte power plant.

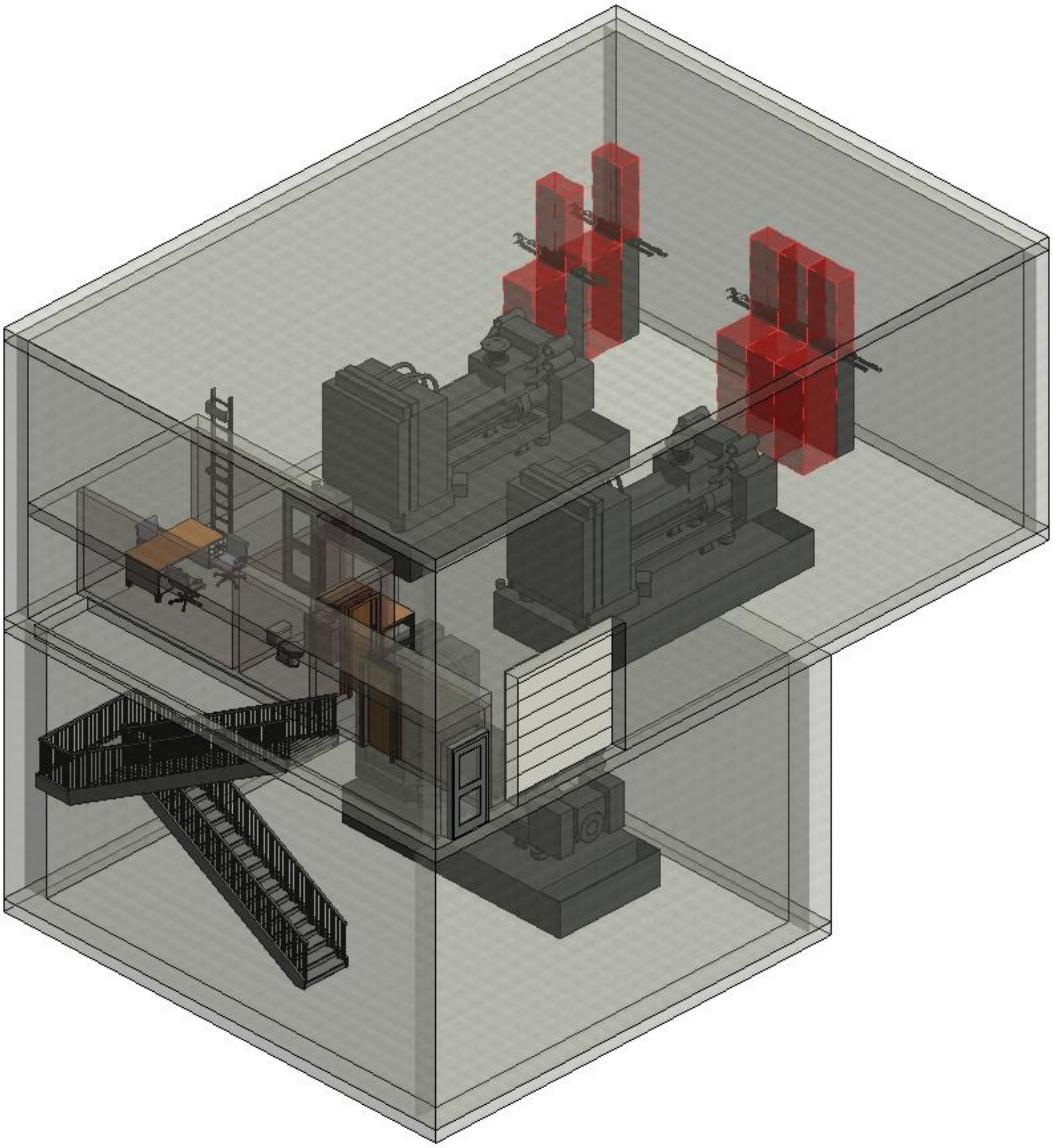

Figure 16: 3D model of Almonte PP building

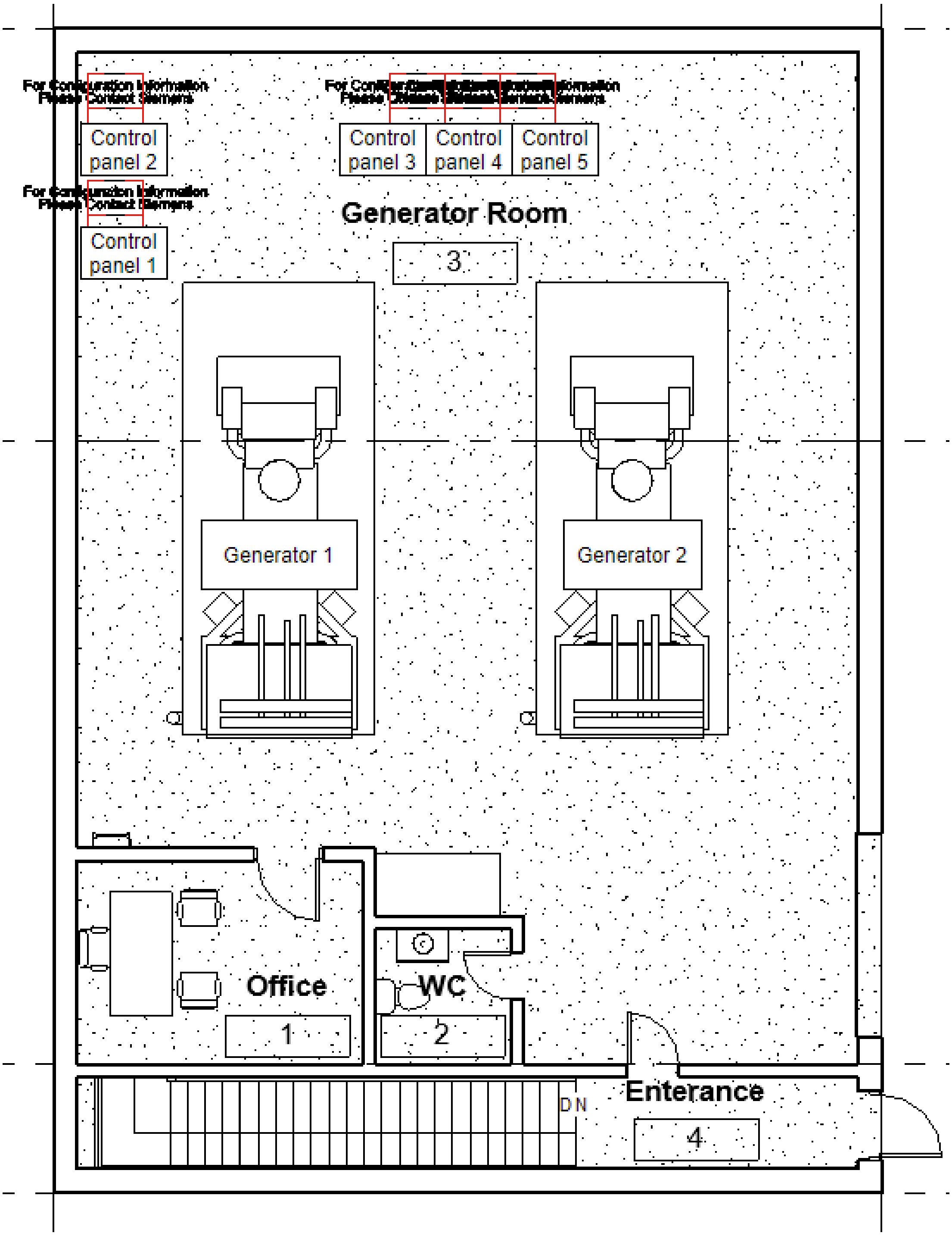

Figure 17: Ground floor plan, Almonte Power Plant.

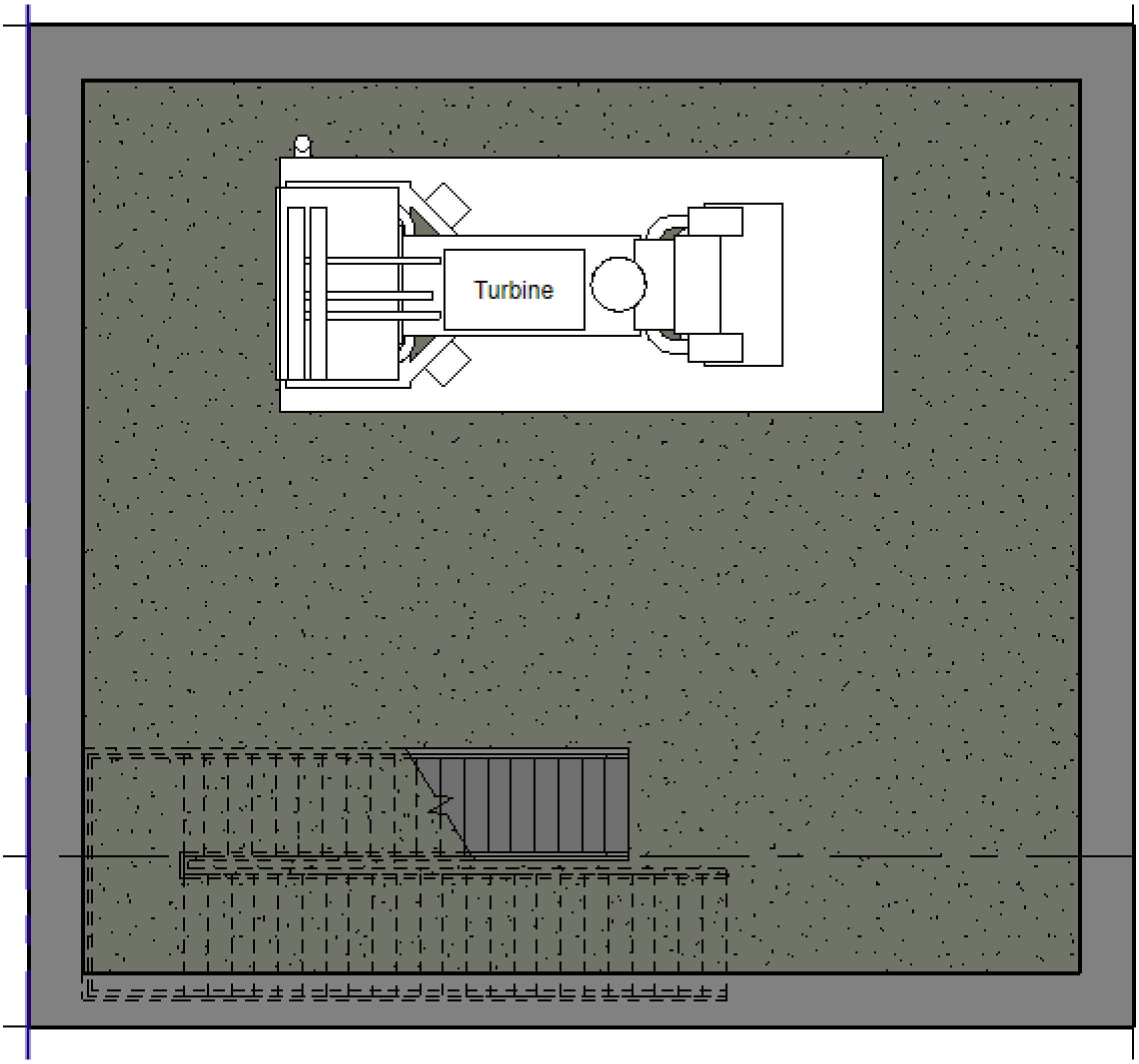

Figure 18: Basement plan, Almonte Power Plant.

Table 2: List of objects in BIM (example)

Family and Type	Category	Room: Name
Door-Exterior-Single	Doors	Office
Single-Flush: 70*200	Doors	WC
Single-Flush: 30" x	Doors	Generator Room
Door-Exterior-Single	Doors	Generator Room
Switchboard-Siemens:	Electrical Equipment	Generator Room
Electrical_Equipment	Electrical Equipment	Generator Room
Electrical_Equipment	Electrical Equipment	Generator Room
T0801 - Workbench, E	Specialty Equipment	Generator Room
Desk: 72" x 36"	Furniture	Office
Switchboard-Siemens:	Electrical Equipment	Generator Room
Switchboard-Siemens:	Electrical Equipment	Generator Room
Switchboard-Siemens:	Electrical Equipment	Generator Room
Switchboard-Siemens:	Electrical Equipment	Generator Room
Toilet-Domestic-3D:	Plumbing Fixtures	WC
Lavatory - Vanity: 3	Plumbing Fixtures	WC
Chair-Task Arms: Cha	Furniture	Office
Chair-Task Arms: Cha	Furniture	Office
Chair-Task Arms: Cha	Furniture	Office
Electrical_Equipment	Electrical Equipment	Turbine Room
Ladder-Access-O'Keef	Specialty Equipment	Generator Room
Door-Overhead-Sectio	Doors	Generator Room

4.3 RISK MODEL

As described in section 2.5, dependency models developed by Atef and Bristow (2019) fall into two categories: spatial and operational. Referring to Table 2 we deduct that elements in the same category can form an operational dependency, though it is not an absolute prerequisite, and elements in the same room can form spatial dependency. Following the spatial dependency criterion, the following network was created (Figure 19).

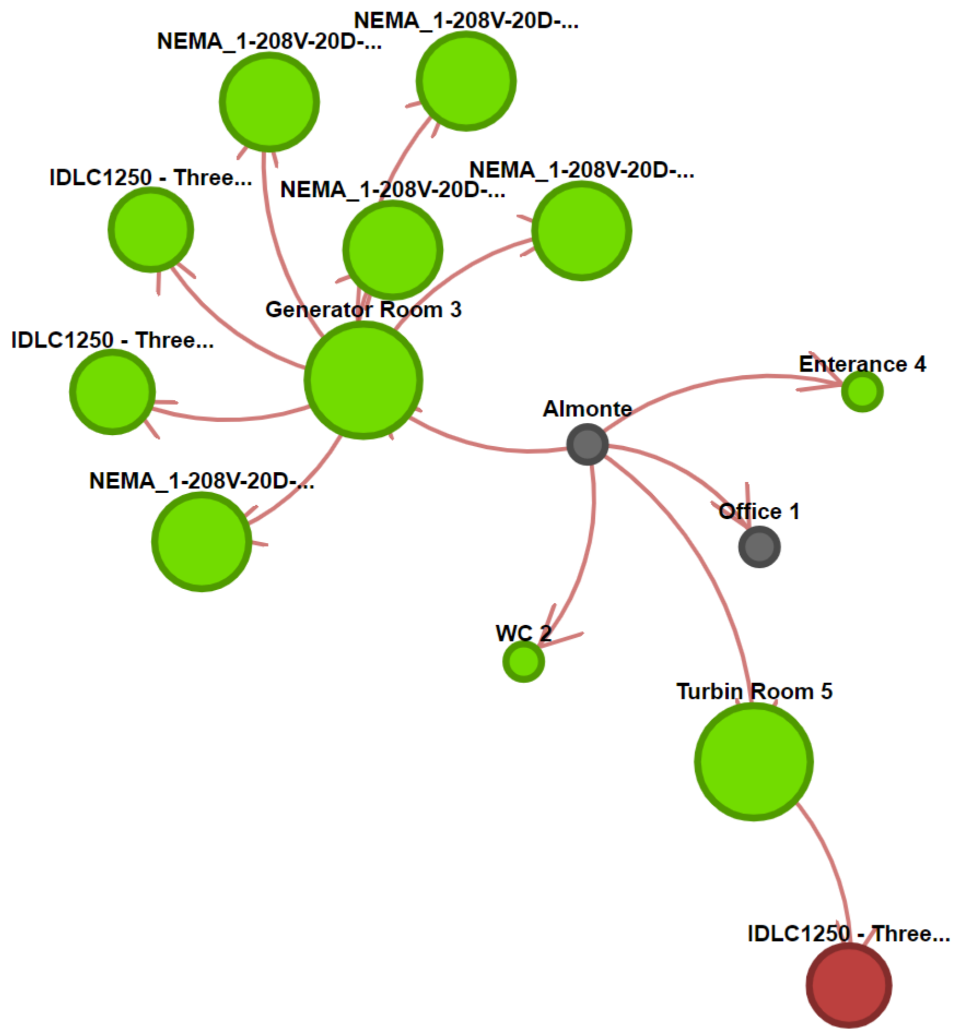

Figure 19: Spatial dependency network.

This particular network contains elements in the "Electrical Equipment" category. Each of the electrical machines is connected to its respective position (i.e. room). Rooms are represented by the nodes Generator Room, WC, Entrance, Office and Turbin room. The hierarchy of nodes is apparent in depiction, the highest level being Almonte (power plant building), followed by various spaces and, finally, individual objects at the lowest level of the hierarchy. From the figure, it is apparent that most of the equipment is located in the generator room, which in turn is located on the ground floor. The ground floor is less susceptible to high flood water levels as it is expected that water will be drained into the basement via a staircase.

The operational network is created by connecting elements of an electrical chain in a logical order. This network is shown in Figure 20. The turbine, located in the turbine room, is directly connected to the generator 1 and generator 2. The dependency between those elements goes both ways, i.e. risk index of any directly affects the risk of failure of the others. The generators are further connected to a number of control units. This relationship is not pronounced, e.g. a short circuit in the network may affect the mentioned panels but with less probability.

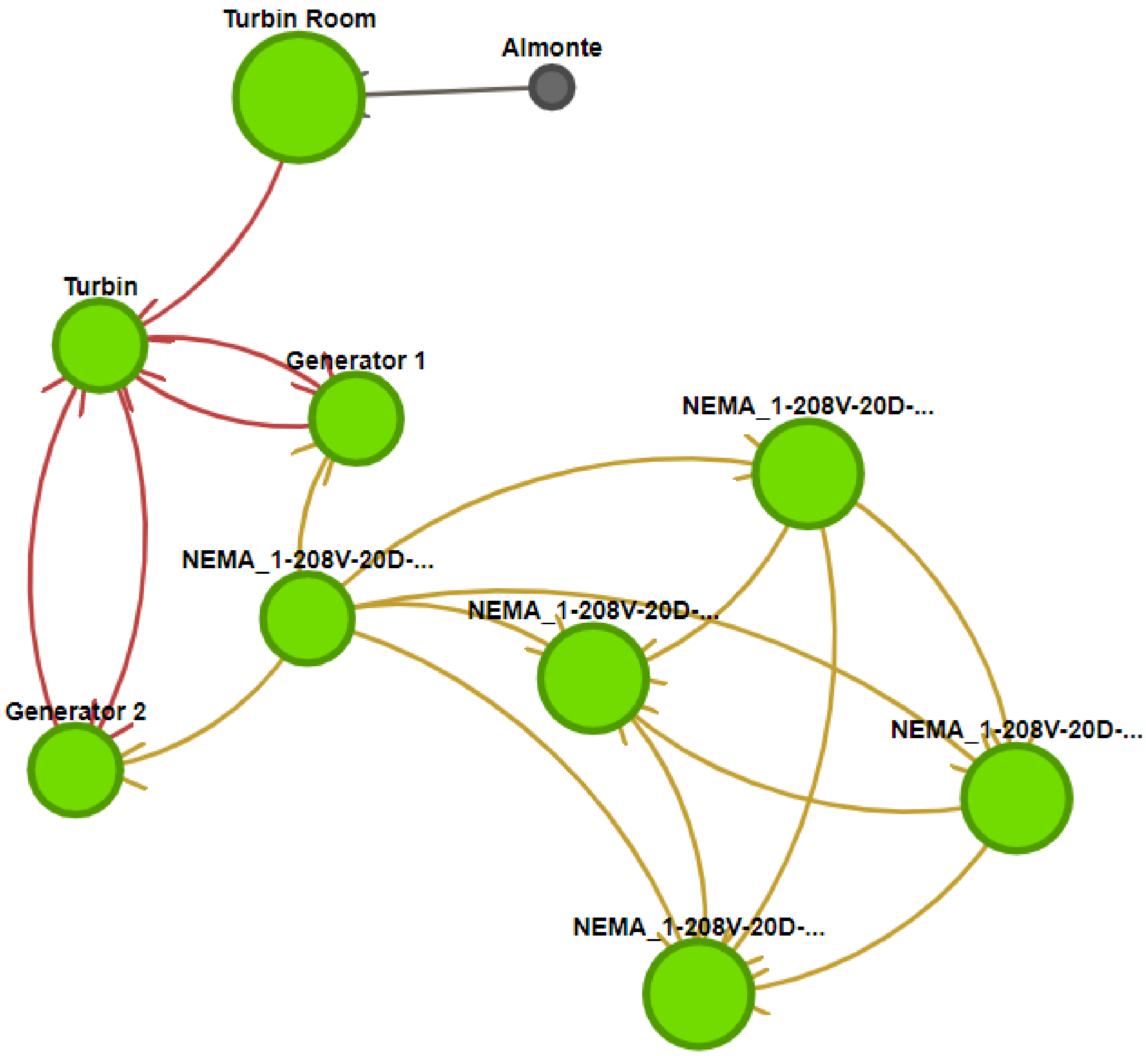

Figure 20: Operational dependency network.

4.4 3D MODEL

The model of Almonte power plant was recreated from BIM using 3DSMax and Unity Editor applications. The obvious difficulty of the process is the loss of information related to objects. To fix this, the model has to be reconnected with data on elements of Table 2. As each element in the Unity scene has his unique identifier it is possible to trace it back to Revit or 3DSMAx and recover information about materials used. At this stage, the materials have to be recreated in the new application, labelled and stored in assets. The materials can be reassigned to the elements manually or using the “material helper tool” developed for this project, which parses the name of the object and decides on appropriate material (e.g. if a name contains keyword “concrete” it corresponds to concrete material). This, in turn, allows the recovering of original visuals of materials as can be seen in Figure 21.

Figure 21: Visual rendering of office furniture.

Section 4.3 introduced a network dependency between elements of the Almonte PP. These interconnections can be visualized with the help of the flowchart presented in chapter

3.2.1. The result of the graphical rendering is given in Figure 22. It is apparent that the turbine is located in the basement and connected via power cables to generators on the ground floor, which in turn connect to control panels. The colour of the spheres moving along the line depicts the strength of the connection. The red colour represents the strongest connection with value 10, and orange, connections of strength 7.

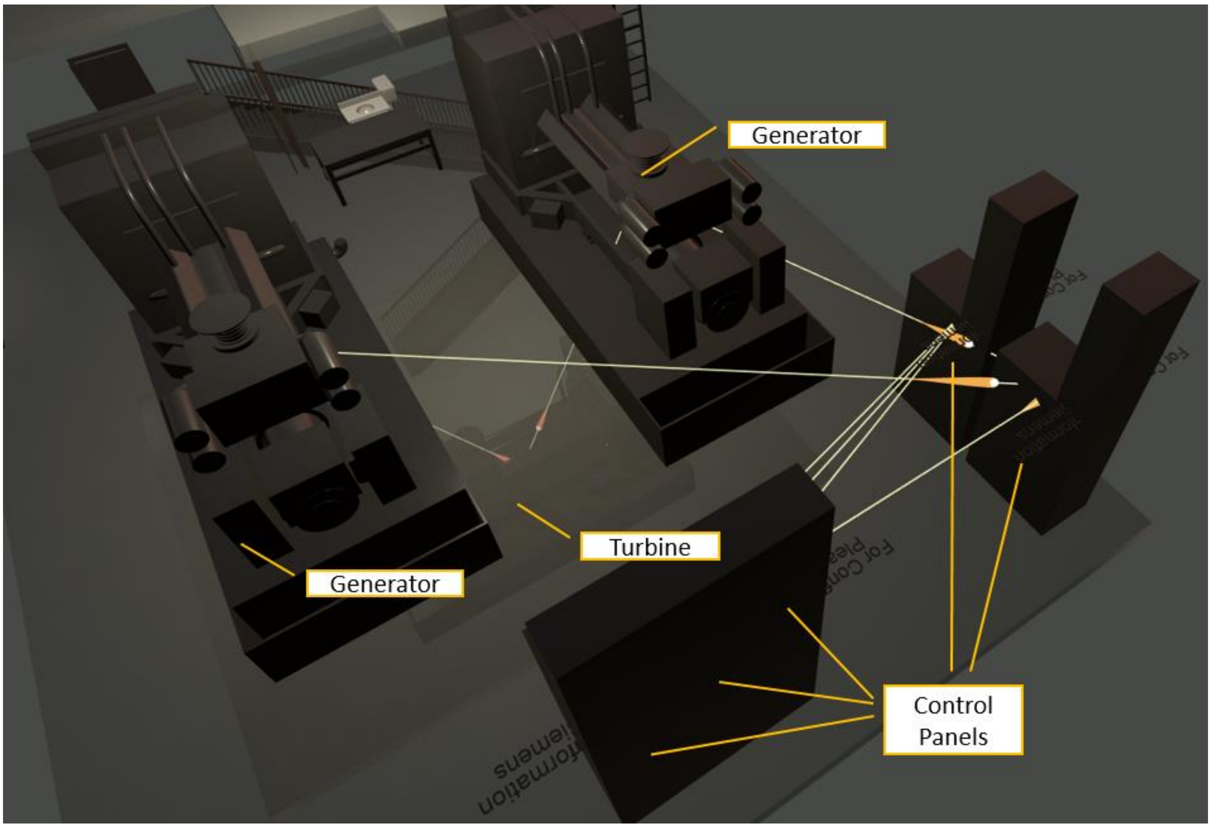

Figure 22: Electrical equipment network depiction in 3D.

The risk score associated with Almonte PP objects is calculated using SiteLogik software and overlaid over the 3D model. Figure 23 depicts an initial state of the system when all of the elements have a low risk score and are coloured green. In Figure 24 risk scores at the final time step of the simulation are presented. Objects coloured red (turbine and 2 generators) have failed by reaching risk score 1.0. The control panels are at significant risk but likely still operational, they are coloured orange. By merging the risk model with the 3D model we portray a context of scenario and progress of failure in the operational chain.

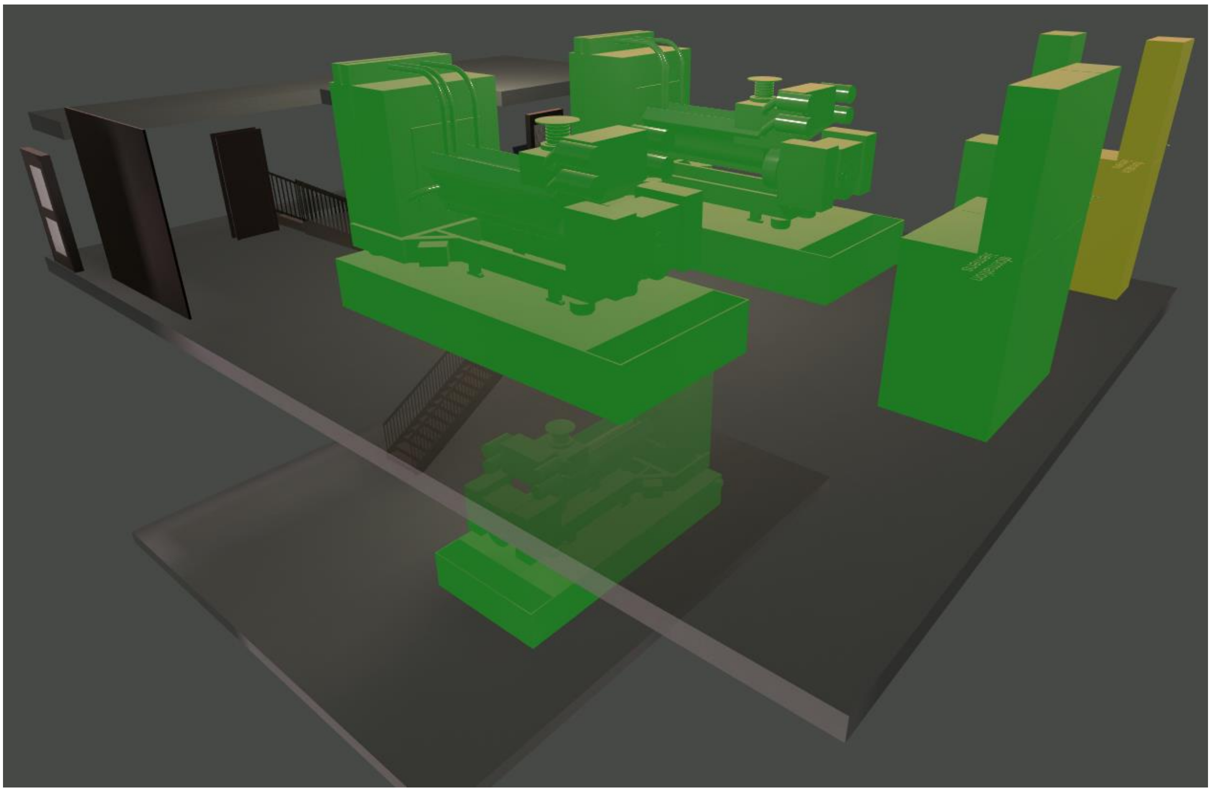

Figure 23: Visualisation of risk scores using colour codes, initial state.

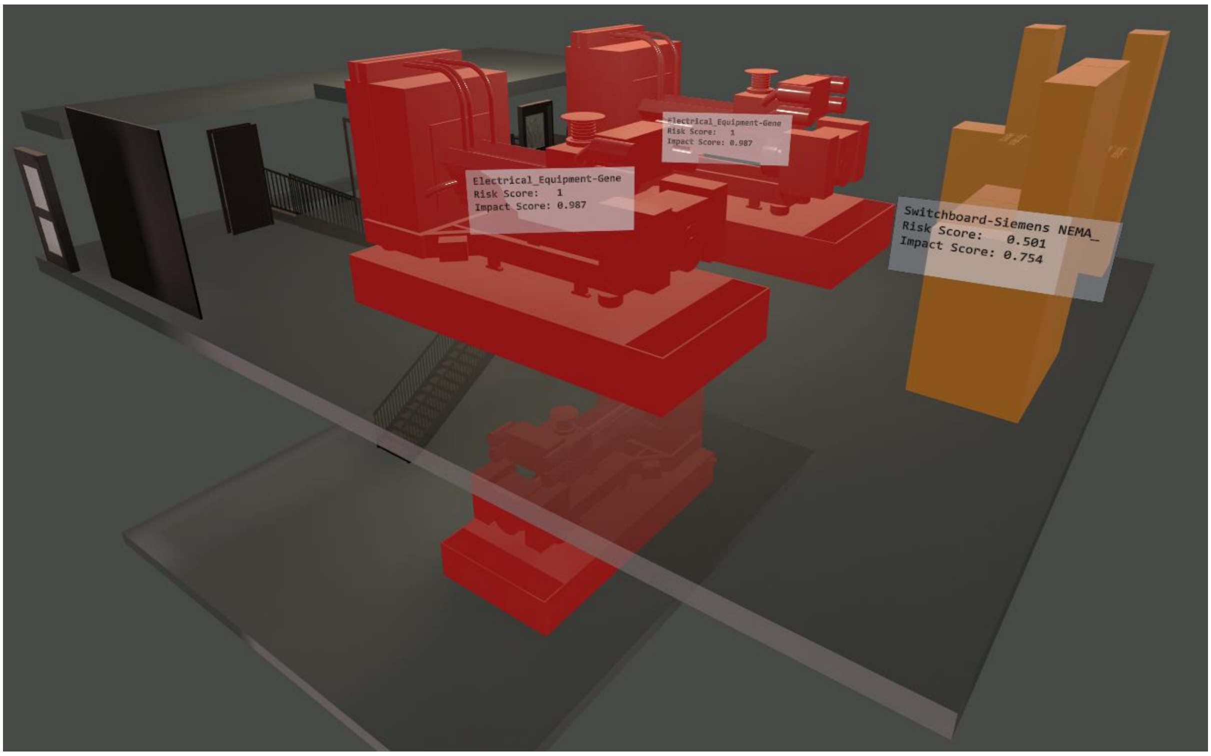

Figure 24: Visualisation of risk scores using colour codes, final state.

4.5 FLOOD MODEL

The base case scenario is a flood of the basement (i.e. turbine room, Figure 25) with free-flowing water from the outside through the main entrance and down the stairs. The water has a steady flow and is uncontaminated. The damage to an electrical turbine is done by coming in direct contact with water volume. The risk score of the turbine is derived from flood depth-damage functions (ref. chapter 2.3).

The depth is calculated using equation:

$$Depth = \frac{Area}{Flow\ Rate \times Time} \tag{5}$$

Area is a known variable, Time and Flow Rate are determined by an engineer for each individual simulation. The net-area (floor area minus area occupied by permanent fixtures) of the turbine room is approximately 87.16 m^2.

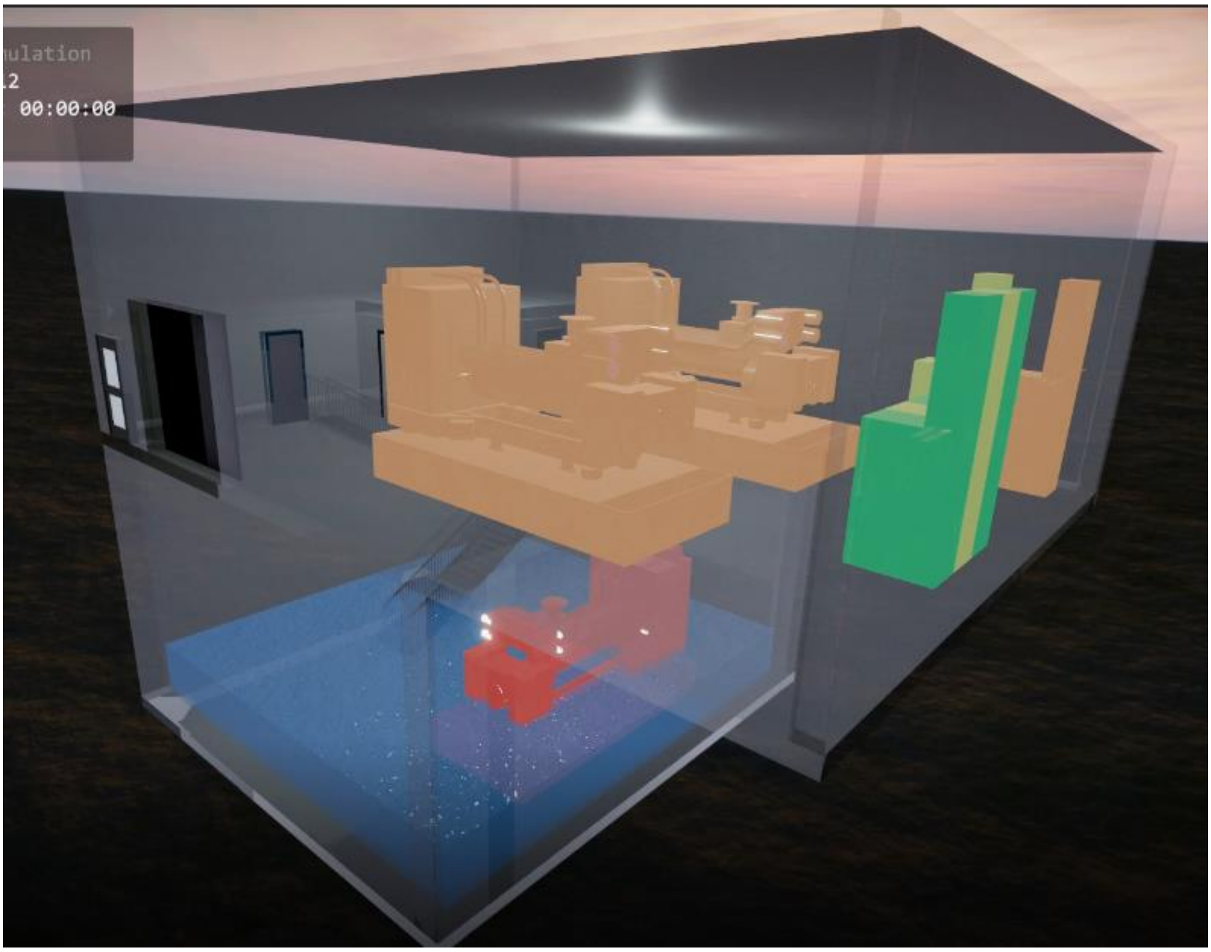

Figure 25: Turbine room flood simulation.

4.6 Risk Analysis Results

Firstly, the basic test of the likelihood of failure of electric machinery depending on the flow rate is performed. The summary of the experimental program is shown in Table 3. Using a series of different flow rates, constant time and floor area value water depth are calculated (fourth column). The flow rates ranging from 0.005-to-0.02 m^3/s are chosen on the assumption that at 4 m depth the turbine will reach a 100% probability of failure in reference to Table 1 and Figure 3. The third column denotes the total time of flood from the moment it starts to reach the maximum water depth. The time of 5 hours is chosen as a reasonable assumption of the period between the start of flood and its termination. Besides, it provides a good range of water depths for chosen flow rates.

Table 3: Basic sensitivity testing program.

Nr.	Flow, m^3/s	Total Time, Hr	Depth, m
1	0.005	5	1.03
2	0.01	5	2.07
3	0.015	5	3.10
4	0.02	5	4.13

Performing analysis using previously developed risk model the results presented in Figure 26 are obtained. The figure depicts failure scores for all elements and flow rates considered in the analysis. Each line connected series corresponds to a separate test. All elements are assigned corresponding numbers on the x-axis.

Summarizing the contents of Figure 26, a total of 4 experiments are performed with the last one at a flow rate of 0.02 m^3/s when a 100% probability of failure of the turbine is reached. The only element in direct contact with the water is the turbine located in the basement. Increasing probabilities of failure of other elements is the result of a chain reaction provided that the failure of the turbine might affect them. Control panel 1 is the least affected by the event since it has no direct connections to the generators and the fewest links to other panels. Control panel number 5 is directly connected to both generators and hence has the highest risk score among all panels. Notably, both generators have the same probability of failure as the turbine due to the tight coupling between these components.

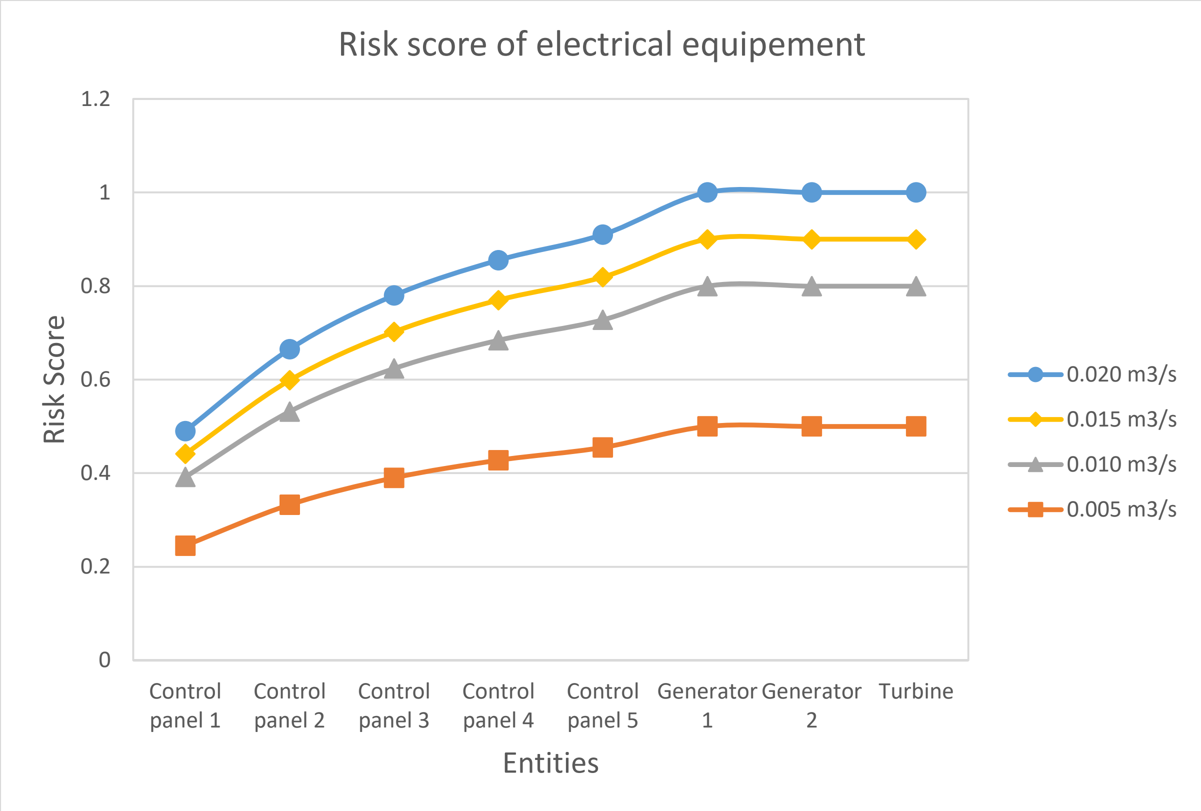

Figure 26: Risk score of electrical equipment after five hours of flooding for different flood flow rates in m^3/s.

Figure 27 shows the growth rate of risk scores with flow rate increases in each consecutive experiment. Risk score curves are given for the turbine, control panel 5 and control panel 1 (from top to bottom). Noticeably, all of the above mentioned have the same growth rate despite the turbine being in direct contact with water. The correlation between the risk score and water flow rate is equal to 0.96. The shape of the graph closely follows the curve of depth-damage functions, which are depicted in Figure 3.The relatively rapid growth of probability of failure between 0.005 m3/s and 0.01 m3/s flow rates is in a good agreement with the shape of depth-damage curves (Figure 3).

In the next experiment dependency between the turbine and generator 1 is modified, which is illustrated in Figure 28. The value of dependency is decreased from 10 to 7, which visibly changed the shape of failure score curve (Figure 29). Predictably, the probability of failure of generator 1 was by 0.7 of the turbine value. Noticeably, it does not affect the score of generator 2, because it is directly connected to the turbine (as seen in Figure 28 generator 1 and 2 are not connected). The ratio between other elements is shown in Figure 30. The risk scores decrease due to weakened dependency in one of the connections in the multi-path analysis (generators 1 and 2 are connected to the panels and affect their probability of failure independently and in parallel).

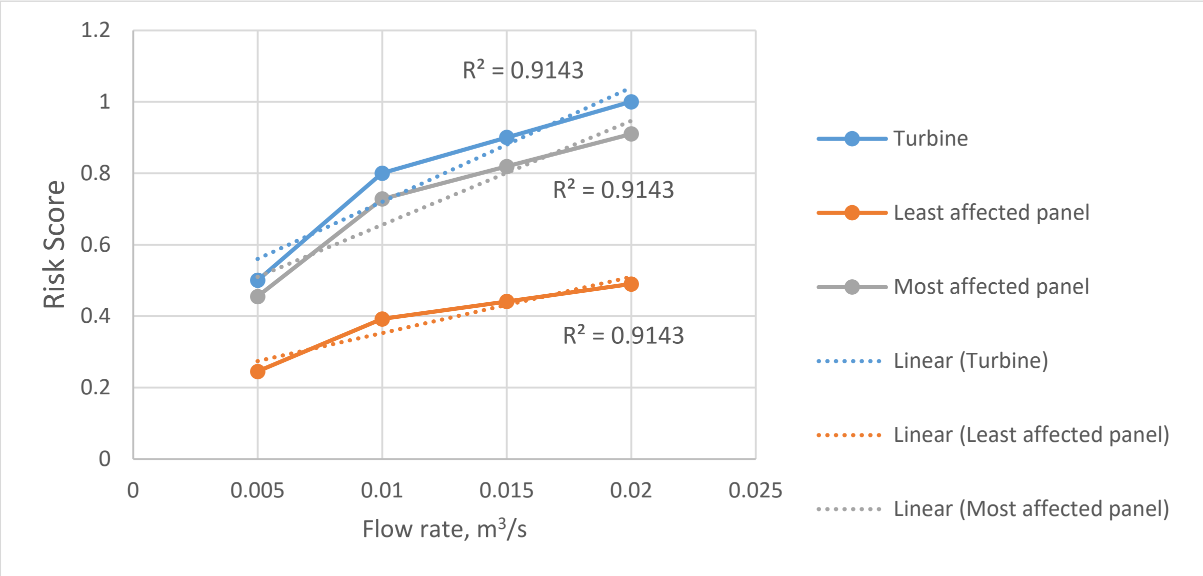

Figure 27: Risk growth to select components due to increased flow rates. Linear regression lines included to better illustrate the larger relative increase in damage from 0.005 to 0.01 m^3/s

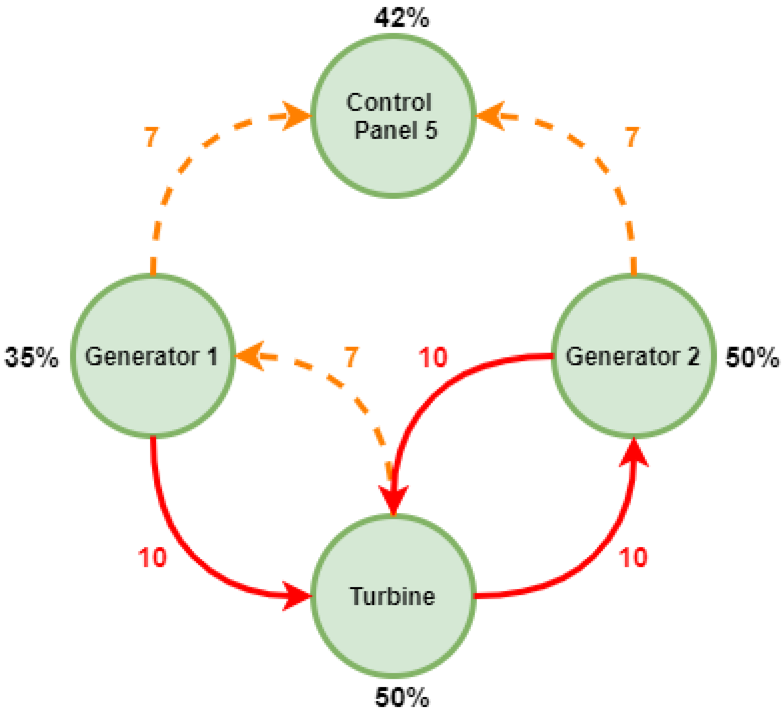

Figure 28: Scheme of modifications to degrees of dependency.

Following the previous discovery of a disproportion of risk score ratio between various elements of the system, a sensitivity analysis is performed. The object of the analysis is the probability of failure of control panel 5. It is measured against the dependency value between the turbine and generator 1, while all the other dependencies remained constant. The results are shown in Figure 31. From the graph, it is apparent that for values of dependency between 7 and 10 the risk increases at constant 0.0105, where Sensitivity = ΔY / ΔX. Nonetheless, for the lower values, the probability of failure remains constantly equal to 0.428. Consequentially, in the case that generator 1 is completely disconnected from the turbine it will not affect risks of other elements below the constant value.

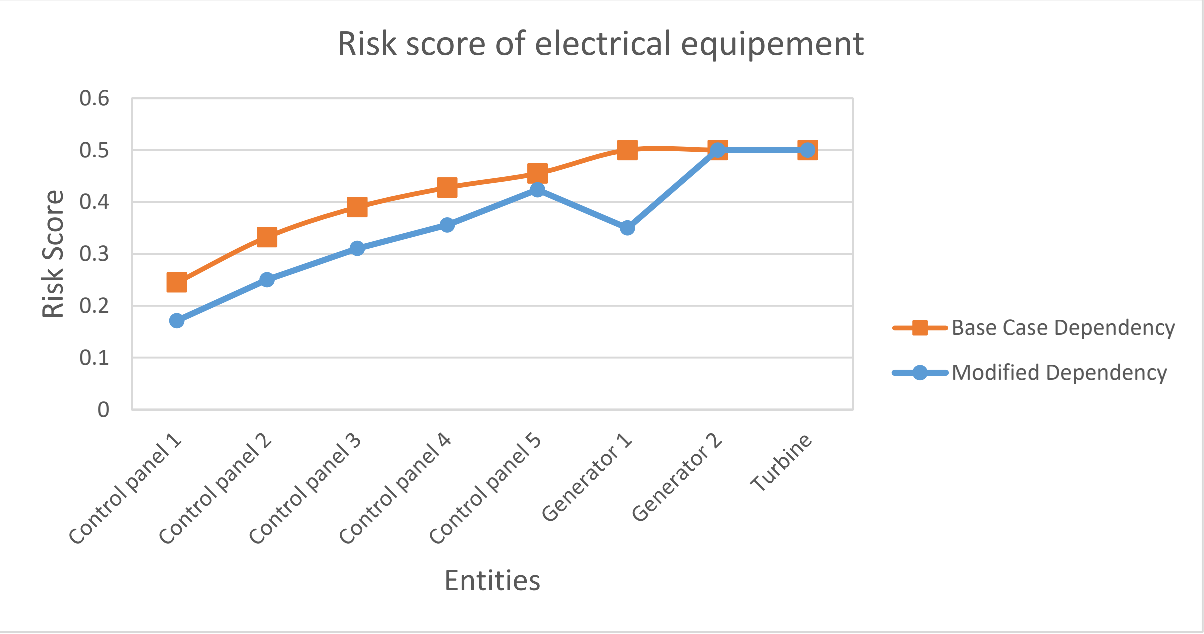

Figure 29: Impact of component dependency on risk scores of electrical components for a flow rate of 0.005 m^3/s.

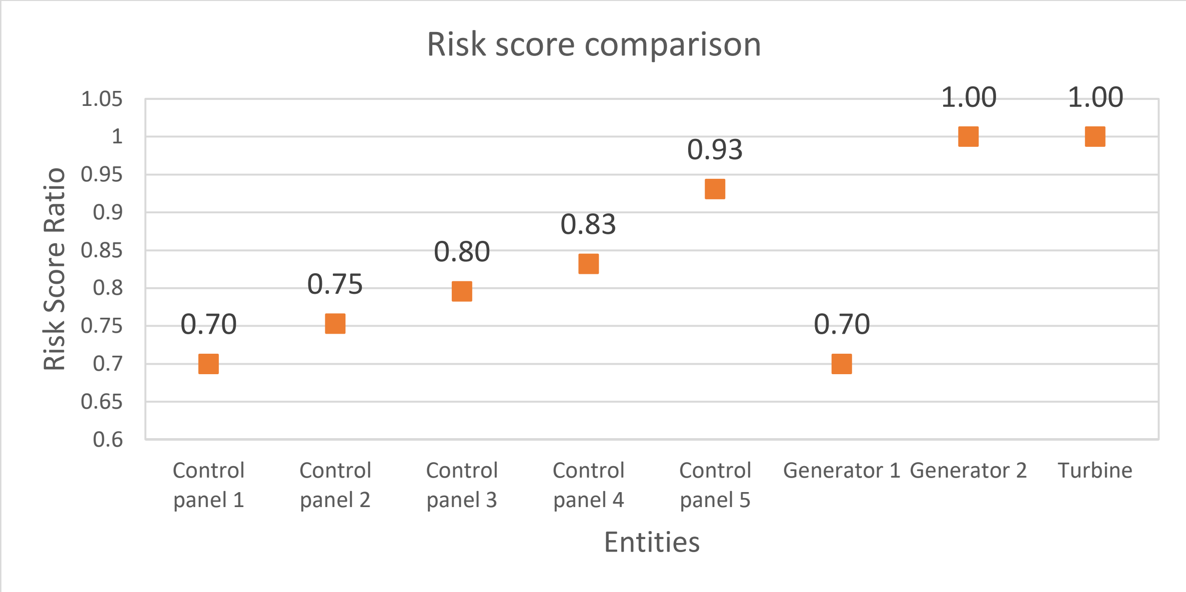

Figure 30: Comparison of risk score of electrical components between cases with different dependency for a flow rate of 0.005 m^3/s.

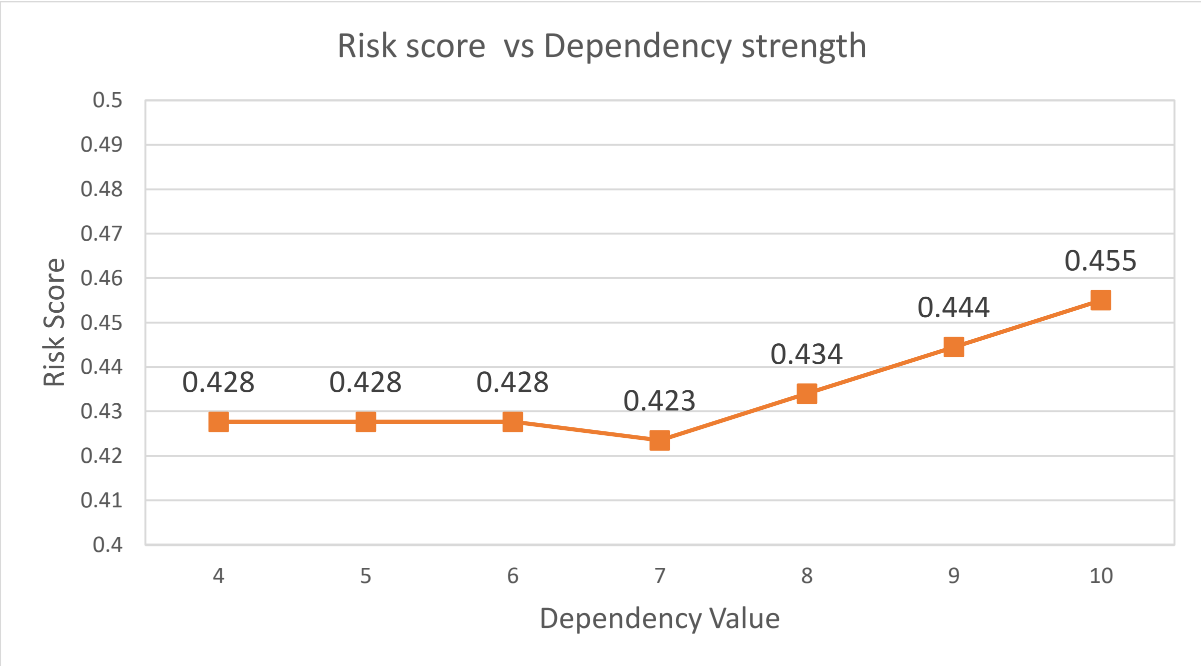

Figure 31: Sensitivity of the risk score of the control panel 5 to the dependency between the turbine and generator 1 for a flow rate of 0.005 m^3/s.

Building on the results of the sensitivity test of a risk score to the dependency degree new experiment is performed. The goal of the experiment is to observe the change of behaviour of the system if the degree of dependency is adjusted to the specific criterion. In this case, the adjustments are based on the amount of "parent" nodes. The observed node risk score depends on the risk score of the "parent" node and degree of dependency between them. Previously in the project, as well as, in this experiment the "OR" rule between concurrent “parent” nodes is assumed, which means that safe operation of at least one is sufficient to support the function of the observed node. Follow this assumption, if the node has several “parents” its degree of dependency on each individual can be lowered. The following rules apply:

- 1 “parent” – no change in the degree of dependency
- 2 “parents” – degree of dependency is reduced by 1 point
- 3 and more “parents” – degree of dependency is reduced by 2 points.

Upon applying this rule, the Table 4 was produced. The turbine appears in the table twice because it has a double-sided connection to the generators, as well as, a connection to the turbine room.

Table 4: Degree of dependency adjustment.

Element	Parents	Initial Value	No of Connections	Value Adjustment	New Value
Control panel 1	Generators	7	2	-1	6
Control panel 2	Control Panels	7	1	0	7
Control panel 3	Control Panels	7	2	-1	6
Control panel 4	Control Panels	7	3	-2	5
Control panel 5	Control Panels	7	4	-2	5
Generator 1	Turbine	10	1	0	10
Generator 2	Turbine	10	1	0	10
Turbine	Turbine room	10	1	None	10
Turbine	Generators	10	2	-1	9

The results of these changes to the model are summarized in the figure. The probability of failure is decreased for all elements preceding control panel 5 in the graph. It can be explained by the decrease in control panel 5 score because it is dependent on generator 1 "OR" generator 2. The risk scores for most of the other elements were decreased for a similar reason, the number of "parents" exceeding 1.

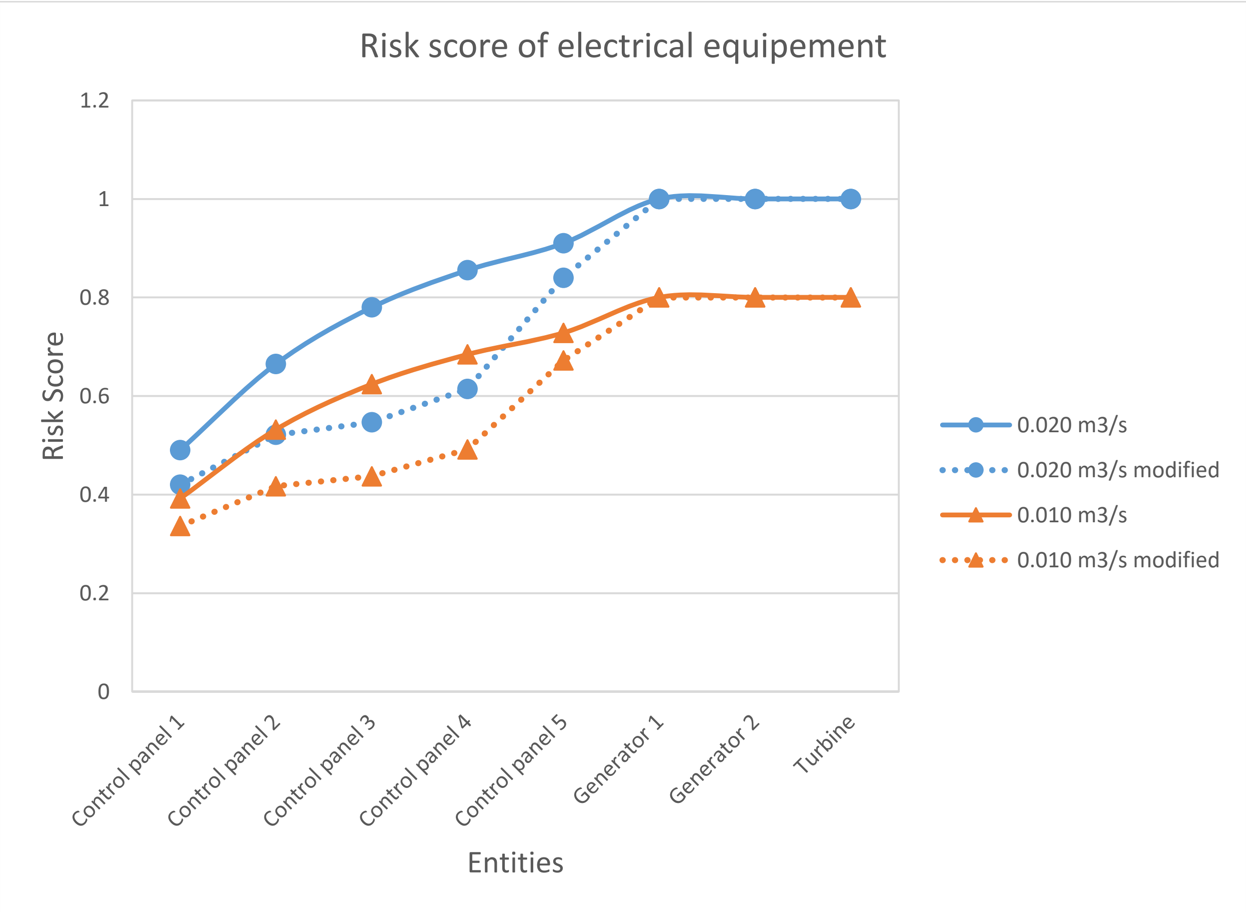

Figure 32: Comparison of baseline risk model and risk model with modified degrees of dependency.

5 DISCUSSION

This project consists of two main parts, the risk visualization software development, and risk model analysis. Each of them has the potential for further improvement. In this chapter, the author wants to discuss possible directions for further work on the subject of this study.

The current work focused on developing a visualization tool for risk assessment with a focus on flood hazards. Further, it considered only one system in the building, electric machines for power generation inside the building. With the work we have done it is possible to extend this narrow field of application to a larger variety of scenarios that include, but are not limited to, earthquakes, explosions, fires, etc. In this project structural, HVAC, security and other assets were excluded from the analysis. The primary reason is that these were the key elements under risk in the hazard scenario.

Working on this project it became apparent to the author that analyzing interconnections between machines in the project is much more comfortable using the 3D model. It is intuitive due to the possibility of interaction with 3D objects (e.g. the connection lines are drawn upon clicking on objects). Charts and tabular data present ample information about the numerical data but lack context, whereas the 3D model intuitively shows the positions in space of every element making vulnerability data intuitive to analyze. In the current state of the developed tool two methods are not interchangeable but instead have to be used side-by-side. Though, with future work, it should be possible to obtain all the necessary info without leaving the application.

The future work on the application includes rigorous user testing to improve the quality of delivery of information via VR. Considering that this project was initiated to accommodate both risk management experts and casual observers (e.g. clients of risk analysis provider), any user testing should also include these audiences. Ideally, the “experts” are chosen from academia, i.e. professors focusing on risk management, as well as, industry specialists. The “client” role is more undetermined and can be filled, for example, by university students if it is not possible to work with facility managers or owners.

This questionnaire is intended to inquire about the potential advantages of the VR model in comparison to charts by answer questions, such as “is the model intuitive?” or “does it convey more information than tabular data?”. The ultimate goal is to present it difficult to grasp numerical information and network graphs in a way accessible to the people unfamiliar with the context of any particular project, i.e. they can grasp the meaning of simulation at a first glance.

The best way to conduct a user test is by asking open-ended questions that do not force a person into predetermined answers. The questioning can be done at the same time as the user is wearing the VR helmet and interacting with a virtual environment. Topics of the questionnaire should cover (1) understanding of the scenario of simulation; (2) feeling of immersion in the scene; (3) accessibility of user interface; (4) possible side-effects (e.g. feeling of nausea); (5) variety of options to control the simulation; (6) suggestions for improvement by the user.

Based on the answers improvements can be implemented in several iterations in a period of several months. In case the level of complexity of simulation is increased in the future it is advised to revisit the user testing routine.

Speaking about the risk model, it is obvious that the concept of degree of dependency is subject to expert judgment and interpretation. The score of 0-to-10 can be translated into the risk score multiplication factor of values 0-to-1.0 in several ways. The current model assumes the linear distribution of multiplication factors, meanwhile, it can be exponential, logarithmic, Poisson, etc. It would be interesting to test the effects of assuming different distributions in future risk models.

One of the complications of the experimental part was the forced assumption of the “OR” rule between several parallel paths. This is an inherent characteristic of the Risklogik model and the strongest path method. The idea is that if at least one path to the element exists, the element gets affected through a chain reaction. With this notion in mind dependency strength values were assigned in Table 4. Though in reality high-tech electronic equipment, such as control panels of a power plant, are connected in a more complicated way, and, occasionally, “AND” rule has to be applied to adequately evaluate the degree of dependency.

Upon exploring various flow rates, it became apparent that considering constant reaction time to the flood, i.e. the time when the water stream is completely cut off by any means, the strength of flow is what determines the amount of damage caused. Speaking in real-world terms, the flow can be controlled through preventive measures. By implementing preparatory protective actions against possible flood the volume of water getting inside the facility can be greatly decreased and risks minimized. From Figure 26 and Figure 27 it is possible to evaluate the actual decrease in risk score with a reduction of water flow to the basement. The ways of protecting the facility from hazards are not explicitly discussed in this work and may be a subject for future study.

The equations used for water volume calculations are rather simplistic and do not include factors like water backflow and drainage, absorption by porous materials and buoyancy of objects not fastened to the floor. The former two require the inclusion of specific elements in the model with total water volume reduction properties upon collision. The latter two have to take into account materials the objects in the scene are made of and upon collision provide adequate reaction. In the current project, we decided not to consider any of them, as it will make the model complicated and require in-depth knowledge of 3D modelling and programing by a casual user, which was not the goal of this endeavour.

6 CONCLUSIONS

This work contributes: (1) a novel method to convert a BIM and an associated risk model into a dynamic virtual reality simulation that can illustrate risks to components due to direct impact and to cascades of effect among building components; (2) A means to apply the VR simulation to the case of flooding induced failure, including a flood simulation using a cellular automaton approach in a confined and irregular three-dimensional space; (3) application of the above methods to a hydroelectric dam to illustrate the experience generated by the simulation in a real facility; and (4) investigation of the relationship of the flood simulation with a proprietary risk analysis engine available from industry.

Firstly, we have converted a BIM project, de-facto modern technology in structural design, into an interactive VR experience. Presently, several commercial projects have achieved this, but none of the open-source software is available for this purpose. It is worth noting that work in this field is a priority in the field of architecture, engineering and construction. Streamlining this process can greatly increase the availability of traditionally engineering specific data to other parties.

A number of VR and 3D simulation tools were developed using the Unity engine to visualize the abstract numerical data on possible hazards to industrial buildings. The advantage of those simulations can be summarized in two concepts: availability for the general public, i.e. the simplicity of visualization of risk using colour code on realistic 3D machines; and time-space context to the simulation, i.e. 3D game-like space of the model that a user can explore in real-time simultaneously to the model progression, in this case, flood propagation through the building. Using the case study of Almonte PP it is illustrated how the possible ways in which failures can evolve for the case of flooding. Noticeably, the risk starts at a low score on directly affected elements (i.e. the turbine) but cascades into the failure of other elements of the electrical system. As expected, the risk increases more quickly, the higher the flood rate.

We have found that in the particular case study of Almonte PP it is theoretically possible to increase the resilience of the system as a whole to a certain extent based on a dependency criterion. Nonetheless, many systems are non-linear with several units working in parallel. It is not always clear if a system can continue to successfully operate if one of the elements in parallel connection fails. In this work, we have shown that generators 1 and 2 can work independently from each other. Nonetheless, it might not be the case on practice, and without stress-testing judgment of safety of the operation can be faulty. It was also observed that at a certain point, that is defined by expert judgment, a ceiling in preventive measures is reached and a further decrease of dependency between connections exemplified in the thesis becomes ineffective.

Summarizing the project, we have managed to use a state of the art risk model and we have transfered it to the virtual reality visualization engine. Of the initial objectives, the flexibility was achieved. The model is not restricted to one specific scenario. The calculation of hydraulics was made separate from the rendering algorithms. As a result, it can be changed according to the complexity of the project. The flood model recognizes collisions with solid objects and water volume responses accordingly. Nonetheless, the transfer of information between water mass and other objects was not realized, i.e. static parts of the building do not have a response upon contact with water. This potentially allows future development of risk simulations not only in traditional media but bringing it up to speed with the emerging technologies of the decade. There is no doubt that virtual and augmented reality technologies will continue rapidly developing, and bridging the gap between civil engineering and modern computer science is a step to progress.

References

Atef, A., Bristow, D., 2019. Capturing The Spatial and Operational Interdependencies Among Building Systems Using Building Information Modelling. Struct. Infrastruct. Eng.

Braley, C., Sandu, A., 2010. Fluid Simulation for Computer Graphics: A Tutorial in Grid Based and Particle Based Methods.

Brown, T., 2008. Design Thinking. Harward Business School.

Brunner, E., Giroux, J., 2009. Examining Resilience. A concept to Improve Societal Security and Technical Safety, CRN Factsheet. Cent. Secur. Stud. CSS ETH Zür.

Dusold, T.C., 2008. Applications of Narrative to the Engineering Decision Making Process and the Pedagogy of Engineering Education (Thesis). Iowa State University.

Eppler, M.J., Aeschimann, M., 2009. A Systematic Framework for Risk Visualization in Risk Management and Communication. Risk Manage. 11.

Eppler, M.J., Aeschimann, M., 2008. Envisioning Risk. A Systematic Framework for Risk Visualization in Risk Management and Communication.

FLOODLIST NEWS, 2019. Canada – Thousands Evacuated After Rivers Flood in Quebec, New Brunswick and Ontario.

Genutis, D., 2013. Testing Flood-Damaged Electrical Equipment. EC&M.

Gesner, T., 2019. "I have never seen the water this high": Residents in Beckwith, Mississippi Mills and Carleton Place Battle Flooding. InsideOttawaValley.com.

Grant, R., 2018. To Help the Analysts Avoid Problems, in: Data Visualization: Charts, Maps, and Interactive Graphics. CRC Press.

Hilfert, T., König, M., 2015. Low-Cost Virtual Reality Environment for Engineering and Construction, in: Proceedings of the 32nd ISARC, Oulu, Finland.

Hu-Chen Liu, Long Liu, Nan Liu, 2013. Risk Evaluation Approaches in Failure Mode and Effects Analysis: A Literature Review. Expert Syst. Appl. 40.

Huizinga, J., de Moel, H., Szewczyk, W., 2017. Global flood depth-damage functions. Methodology and the database with guidelines.

IICRC, 2015a. ANSI/IICRC S500 Standard for Professional Water Damage Restoration, 4th ed. USA.

IICRC, 2015b. ANSI/IICRC S520 Standard for Professional Mold Remediation, 3rd ed. USA.

Kumaran, G.S., Rubesh Anad, P.M., Santhi, K.R., 2007. Impact of Augmented Reality (AR) in Civil Engineering. Adv. Mater. Res.

Merz, B., Kreibich, H., Lall, U., 2013. Multi-Variate Flood Damage Assessment: a Tree-Based Data- Mining Approach. Nat. Hazards Earth Syst. Sci. NHESS 1.

MITRE, 2014. Systems Engineering Guide.

Moreno, R., Mayer, R.E., 2002. Learning Science in Virtual Reality Multimedia Environments: Role of Methods and Media. J. Educ. Psychol. 94.

Ni, D., Yew, A.W.W., Ong, S.K., Nee, A.Y.C., 2017. Haptic and Visual Augmented Reality Interface for Programming Welding Robots. Adv. Manuf. 5.

Nickel, P., Kergel, R., Wachholz, T., Proger, E., Lungfiel, A., 2015. Setting-up a Virtual Reality Simulation for Improving OSH in Standardisation of River Locks, in: Conference: 8th International Conference Safety of Industrial Automated Systems – SIAS 2015. Königswinter, Germany.

O'Neil, P., 2017. Building Resilience Through Risk Analysis, NATO Science for Peace and Security Series C: Environmental Security.

O'Neil, P., 2013. Protecting Critical Infrastructure by Identifying Pathways of Exposure to Risk. Timereview.Ca.

Pistrika, A., Tsakiris, G., Nalbantis, I., 2014. Flood Depth-Damage Functions for Built Environment. Environ. Process.

Puschmann, P., Horlitz, T., Wittstock, V., Schütz, A., 2016. Risk Analysis (Assessment) Using Virtual Reality Technology - Effects of Subjective Experience: An Experimental Study, in: Proceedings of 26th CIRP Design Conference, Stockholm, Sweden.

Reed, C., 2018. 7 Things You Can Do to Overcome VR Motion Sickness. Upload.

Riccitiello, J., 2019. Unity Acquires Vivox, which Powers Voice Chat in Fortnite and League of Legends.

Roth, F., 2012. Visualizing Risk: The Use of Graphical Elements in Risk Analysis and Communications. Cent. Secur. Stud. CSS ETH Zür.

Rüppel, U., Schatz, K., 2011. Designing a BIM-based Serious Game for Fire Safety Evacuation Simulations. Adv. Eng. Inform.

Sampaio, A.Z., Ferreira, M.M., Rosário, D.P., Martins, O.P., 2010. 3D and VR Models in Civil Engineering Education: Construction, Rehabilitation and Maintenance. Autom. Constr. 19.

Sampaio, A.Z., Henriques, P.G., 2008. Visual Simulation of Civil Engineering Activities: Didactic Virtual Models., in: Proceedings of 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision.

Sharma, A., Singh, S., Bajpai, P., Khatter, K., 2018. Virtual Reality: Blessings and Risk Assessment. Indian J. Sci. Technol.

Shaugnessy, R., Morey, P., Cole, E., 1999. Bioaerosols: Assessment and Control. Presented at the Conference of Govermental Industrial Hygienists.

Stanney, K., Kingdon, K.S., Graeber, D., Kennedy, R.S., 2002. Human Performance in Immersive Virtual Environments: Effect of Exposure Duration, User Control, and Scene Complexity. Hum. Perform. 15.

Sutcliffe, A.G., Deoul Kaur, K., 2000. Evaluating the Usability of Virtual Reality User Interfaces. Behav. Inf. Technol. 19.

Thieken, A., Muller, M., Kreibich, H., Merz, B., 2005. Flood ¨Damage and Influencing Factors: New insights From the August 2002 Flood in Germany. Water Resour Res.

USACE, 2015. Physical Depth-Damage Function Summary Report. USA.

USACE, 2006. Depth-Damage Relationships for Structures, Contents, and Vehicles and Content-to-Structure Value Ratios (CSVR) in Support of the Donaldsonville to The Gulf, Louisiana, Feasibility Study.

USACE, 2003. Economic Guidance Memorandum (EGM) 04-01, Generic Depth-Damage Relationships for Residential Structures with Basements.

Webley, P., Lea, S., 1997. Topic 3: Path analysis, lecture notes, PSY6003 Advanced statistics: Multivariate analysis II: Manifest variables analyses, University of Exeter.

Welch, R.B., 1999. How Can We Determine if the Sense of Presence Affects Task Performance? PRESENCE Virtual Augment. Real. 8.

Wind, H.G., Nierop, T.M., de Blois, C.J., de Kok, J.L., 1999. Analysis of flood damages from the 1993 and 1995 Meuse floods. Water Resour Res.

Woksepp, S., Olofsson, T., 2008. Credibility and Applicability of Virtual Reality Models in Design and Construction. Adv. Eng. Inform.

Xinyu Jiang, Hirokazu Tatano, Tomoharu Hori, 2013. A Methodology for Spatial Flood Risk Assessment Taking Account of the Spatial–Temporal Characteristics of Rainfall. J. Integr. Disaster Risk Manag. 1.

Zhai, G., Fukuzono, T., Ikeda, S., 2005. Modeling Flood Damage: Case of Tokai Flood 2000. Water Resour As.

Appendix A

Almonte simulation model ¶

In [13]:
```
import sys
sys.path.append(r'C:\Users\Vitali\Documents\sitelogik-simulation-engine-maste
r')
import os
os.chdir(r'C:\Users\Vitali\Documents\sitelogik-simulation-engine-master')


from sim_new import *
from ipysheet import from_dataframe, to_dataframe
```

Get available models

In [87]:
```
get_models()
```

```
Models:
AI-2.romdl
AI.romdl
almonte_testing.romdl
big_building.romdl
sample_model.romdl
```

Get available scenarios for one of these models

In [88]:
```
get_scenarios("AI")
```

```
No Scenarios Found
```

Create a new scenario for this model

In [89]:
```
create_scenario(base_model="AI",
                scenario_name="AI-2_scenario_1")
```

```
Successfully created scenario in "scenarios/AI-2_scenario_1.romdl"
```

List and edit entities

Define our entity save function

In [90]:
```
def save_entity_sheet(change):
    print("saving sheet")
    df = to_dataframe(sheet)

    #Add unused columns back
    df['null'] = np.nan
    df['is_building'] = "false"
    df.loc[df['parent_id'] == -1, "is_building"] = True

    df = df[["entity_id", "parent_id", "name", "null", "impact", "failure", "i
s_building"]]#Reorder sheet

    save_entity_frame("AI-2_scenario_1", df)
```

Render the sheet

In [91]:
```
df = get_entities("AI-2_scenario_1")
df.drop(columns=["null", "is_building"], inplace=True) #Drop unused columns
sheet = from_dataframe(df)


#Watch for changes and save to disk
sheet.cells[3].observe(save_entity_sheet)
sheet.cells[4].observe(save_entity_sheet)
sheet
```

	entity_id	parent_id	name	null	impact	failure	is_building
0	505050505	-1	Almonte	NaN	0	0	False
1	366871	505050505	Turbin Room	NaN	10	0	False
2	371460	366871	Turbin	NaN	5	0	False
3	356455	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	5	0	False
4	366566	366862	Generator 1	NaN	5	0	False
5	366803	366862	Generator 2	NaN	5	0	False
6	367565	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False
7	367628	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False
8	367664	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False
9	367698	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-91-f7c1e243ad0e> in <module>
      1 df = get_entities("AI-2_scenario_1")
----> 2 df.drop(columns=["null", "is_building"], inplace=True) #Drop unused c
olumns
      3 sheet = from_dataframe(df)
      4
      5

AttributeError: module 'IPython.display' has no attribute 'drop'
```

List and edit relationships

Define our relationship save function

In [92]:
```
def save_relationship_sheet(change):
    df = to_dataframe(sheet)

    #Add unused columns back
    df['#'] = range(1,df.shape[0]+1)
    df['null'] = np.nan

    df = df[["#", "From Entity ID", "To Entity ID", "null", "degree", "disable
d"]]#Reorder sheet leaving out unused columns

    display(df)

    save_relation_frame("AI-2_scenario_1", df)
```

Render the sheet

In [131]:
```
df = get_relationships("AI-2_scenario_1")
entities = get_entities("AI-2_scenario_1")

df.drop(columns=["#", "null"], inplace=True) #Drop unused columns
df.rename(columns={'From Entity': 'From Entity ID', 'To Entity': 'To Entity I
D'}, inplace=True)

#Add columns for entity names
df['From Name'] = df['From Entity ID'].map(entities.set_index('entity_id')['na
me'])
df['To Name'] = df['To Entity ID'].map(entities.set_index('entity_id')['name'
])

df=df[['From Name', 'From Entity ID', 'To Name', 'To Entity ID', 'degree', 'di
sabled']] #Reorder sheet

sheet = from_dataframe(df)
sheet.cells[4].observe(save_relationship_sheet)
sheet.cells[5].observe(save_relationship_sheet)
sheet
```

	ID1	ID2	null	degree	null_2
0	366871	371460	NaN	10	False
1	371460	366566	NaN	0	False
2	371460	366803	NaN	10	False
3	356455	367565	NaN	7	False
4	356455	367628	NaN	7	False
5	356455	367664	NaN	7	False
6	356455	367698	NaN	7	False
7	367565	367628	NaN	7	False
8	367565	367664	NaN	7	False
9	367565	367698	NaN	7	False
10	367628	367664	NaN	7	False
11	367628	367698	NaN	7	False
12	367664	367698	NaN	7	False
13	366566	371460	NaN	10	False
14	366803	371460	NaN	10	False
15	366566	356455	NaN	7	False
16	366803	356455	NaN	7	False

	entity_id	parent_id	name	null	impact	failure	is_building
0	505050505	-1	Almonte	NaN	0	0	False
1	366871	505050505	Turbin Room	NaN	10	0	False
2	371460	366871	Turbin	NaN	5	5	False
3	356455	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	5	0	False
4	366566	366862	Generator 1	NaN	5	0	False
5	366803	366862	Generator 2	NaN	5	0	False
6	367565	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False
7	367628	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False
8	367664	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False
9	367698	366862	NEMA_1-208V-20D-32W-90H-Electrical Equipment	NaN	7	0	False

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-131-146e80bf4881> in <module>
      2 entities = get_entities("AI-2_scenario_1")
      3 
----> 4 df.drop(columns=["#", "null"], inplace=True) #Drop unused columns
      5 df.rename(columns={'From Entity': 'From Entity ID', 'To Entity': 'To
 Entity ID'}, inplace=True)
      6 

AttributeError: 'NoneType' object has no attribute 'drop'
```

~~Modify some failure and impact values~~ Optional, this is done in the spreadsheet above

In [119]:

```
modify_entity_failure(scenario_name="AI-2_scenario_1",
                      entity_id=371460,
                      failure=5)

modify_entity_failure(scenario_name="AI-2_scenario_1",
                      entity_id=366803,
                      failure=0)
```

```
Successfully modified failure for AI-2_scenario_1.romdl - 371460 from 5 to 5
Successfully modified failure for AI-2_scenario_1.romdl - 366803 from 0 to 0
```

~~Modify some relationship levels~~ Optional, this is done in the spreadsheet above

In [130]:

```
modify_entity_relationship(scenario_name="AI-2_scenario_1",
                           entity1_id=371460,
                           entity2_id=366566,
                           degree=0)
```

```
Successfully modified degree for AI-2_scenario_1.romdl - 371460->366566 from
4 to 0
```

Run a simulation on the current scenario

In [132]:

```
run_simulation(scenario_name="AI-2_scenario_1")
```

```
Simulations complete. Results available in:
processed_simulations/processed_AI-2_scenario_1.csv
```

Viewing Simulation Results

All processed simulations are available in processed_simulations/

Appendix B

...ect Risk in VR\Assets\Scripts\Temporary\VoxelGenerator.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[RequireComponent(typeof(MeshFilter), typeof(MeshRenderer))]
public class VoxelGenerator : MonoBehaviour {

    // Version 1.1

    //Voxel Codes
    // 1  == grid cell is active
    // 0  == grid cell is empty
    // -1 == grid cell is blocked

    //Voxel grids
    int[,,] currentGrid;
    int[,,] nextGrid;
    int[,,] controlVolumeGrid;
    int[,,] antiGrid;

    public Camera mainCamera;

    public int width; //Grid width x
    public int height; //Grid height y
    public float particleScale = 1.0f; //Size of one water particle

    public GameObject waterParticle; //particle to clone
    GameObject particle; //particle to spawn
    GameObject pool; //parent of all water particles
    public Vector3 origin; // position of the first particle

    Queue<GameObject> objectPool; //Stores particles, invoked on Start()

    float nextVolume; // total next volume of all water particles in cubic meters
    const float layerZ = 0.1f; // water level height in meters

    public float flow; //flow in cubic meters per second
    public float timeScale; //to speed up time
    public int maxParticleCount = 20000; //water particle number in Queue

    bool alternate; //determines if main or alternative celular automate search algorhitm is used

    public int count = 0; //particle count
    int prevCount = -1;
    public float volume; //current volume flow * time in cubic meters
    public float spillArea; //spill area

    bool mouseClicked; // used to start Simulation

    //Timers
    float startTime = 0;
```

```
    public float elapsedTime;

    public bool partcileCountCheck = false;

    #region Singleton

    public static VoxelGenerator Instance;

    public Camera objectToFollow;
    public Camera fallbackObjectToFollow;

    private void Awake()
    {
        Instance = this;
    }

    #endregion


    // Use this for initialization
    void OnEnable ()
    {
        ResetVariables();
    }

    // Update is called once per frame
    /*
    void Update () {


        if (startTime != 0)
        {
            elapsedTime = Time.time - startTime;
        }

        if (Input.GetMouseButtonDown(0))
        {
            ResetVariables();

            PlayerControls controls = mainCamera.GetComponent<PlayerControls>();
            controls.PointAndClick();
            origin = controls.origin;

            mouseClicked = true;

            spillArea = 0;

            CreateObjectPool();
            CreateBasicGrid();

            initializeVertices();

```

```
                CreateShape(currentGrid);


                SpawnParticles();


                Volume();


            }

            if (mouseClicked) // stops Simulation
            {
                if (Input.GetMouseButtonDown(2))
                {
                    mouseClicked = false;
                }

                CalculateFlow();

                checkControlVolume();

                if (volume > nextVolume)
                {
                    RunSimulation();
                }

            }
        }
        */
        //Main body of the script
        public void SetSimulationParameters()
        {
            ResetVariables();

            CreateObjectPool();
            CreateBasicGrid();

            initializeVertices();
        }

        public int[,,] RunSimulation()
        {

            //Terimnates water horizontal propagation
            if (prevCount == count)
            {
                partcileCountCheck = true;
                return currentGrid;
            }

            AdjustGrid();
```

```

        GenerateNextGrid();

        CreateShape(currentGrid);
        UpdateMesh();

        currentGrid = GridSubstract(nextGrid, currentGrid);


        prevCount = count;

        SpawnParticles();


        currentGrid = nextGrid;

        return null;
    }
    #region VolumeControl
    //Calculates water volume treshhold for the next step in voxel grid
    void checkControlVolume()
    {
        int[,,] reserveGrid = currentGrid;
        AdjustGrid();
        GenerateControlGrid();
        Volume();
        currentGrid = reserveGrid;
    }

    //Calculates theoretical water volume using computation hydro-dinamics
    public void CalculateFlow()
    {
        volume += flow * timeScale * Time.deltaTime;
    }

    //Calculate next step total volume
    void Volume()
    {
        nextVolume = 0;
        for (int x = 0; x < width; x++)
        {
            for (int y = 0; y < height; y++)
            {
                if (controlVolumeGrid[x, y, 0] == 1)
                {
                    nextVolume += particleScale * particleScale * layerZ;
                }
            }
        }
        //Debug.Log(nextVolume);
    }
    #endregion
```

```
    #region ParticleGenerator
    //Creates New Grid
    void CreateBasicGrid()
    {

        if (height % 2 == 0)        {            height++;        }
        if (width % 2 == 0) { width++; }

        currentGrid = new int[width, height, 1];
        currentGrid.Equals(0);

        antiGrid = new int[width, height, 1];
        currentGrid.Equals(0);

        currentGrid[(width - 1) / 2, (height - 1) / 2, 0] = 1;
    }

    //Generates next grid
    void GenerateNextGrid()
    {

        nextGrid = new int[width, height, 1];
        nextGrid.Equals(0);

        for (int x = 0; x < width; x++)
        {
            for (int y = 0; y < height; y++)
            {
                if (currentGrid[x, y, 0] != -1)
                {
                    bool check = CheckNeighborhood(x, y);
                    if (check)
                    {
                        nextGrid[x, y, 0] = 1;

                    }
                }
                else { currentGrid[x, y, 0] = -1; }

            }
        }
        alternate = !alternate;
    }

    //Generates control grid
    void GenerateControlGrid()
    {


        controlVolumeGrid = new int[width, height, 1];
        controlVolumeGrid.Equals(0);

```

```
        for (int x = 0; x < width; x++)
        {
            for (int y = 0; y < height; y++)
            {
                if (currentGrid[x, y, 0] != -1)
                {
                    bool check = CheckNeighborhood(x, y);
                    if (check)
                    {
                        controlVolumeGrid[x, y, 0] = 1;

                    }
                }

            }
        }
    }
    //Used to generate invisible water particles that detect collision
    void SpawnParticles()
    {
        if (currentGrid != null)
        {
            for (int x = 0; x < width; x++)
            {
                for (int y = 0; y < height; y++)
                {
                    Vector3 pos = new Vector3(origin[0] + (-width / 2 + x) * particleScale, origin[1], origin[2] + (-height / 2 + y) * particleScale);

                    if (currentGrid[x, y, 0] == 1)
                    {
                        particle = SpawnFromPool(pos, Quaternion.identity);
                        count++; spillArea = count * particleScale * particleScale;

                        float updatedScale = particleScale - 0.0001f; //prevents unwanted triggering
                        particle.transform.localScale = new Vector3(updatedScale, updatedScale, updatedScale);

                        particle.GetComponent<CollisionBehaviour>().voxelPosition = new int[] { x, y, 0 };
                    }
                }
            }
        }
    }

    //Checks if the grid cell will be equal to 1 in the nextGrid
    bool CheckNeighborhood(int gridX, int gridY)
    {
```

```
        bool hit = false;
        for (int neighbourX = gridX - 1; neighbourX <= gridX + 1; neighbourX++)
        {
            for (int neighbourY = gridY - 1; neighbourY <= gridY + 1; neighbourY++)
            {
                if (neighbourX >= 0 && neighbourX < width && neighbourY >= 0 && neighbourY < height)
                {
                    if (alternate) //alternative search algorhitm
                    {
                        if (currentGrid[neighbourX, neighbourY, 0] > 0)
                        {
                            hit = true;
                        }
                    }
                    else //prime search algorhitm
                    {
                        if (neighbourX == gridX || neighbourY == gridY)
                        {
                            if (currentGrid[neighbourX, neighbourY, 0] > 0)
                            {
                                hit = true;
                            }
                        }
                    }
                }
            }
        }
        return hit;
    }

    //Modifies current grid to account for collisions (sets values to -1)
    void AdjustGrid()
    {
        for (int x = 0; x < width; x++)
        {
            for (int y = 0; y < height; y++)
            {
                if (antiGrid[x,y,0] == -1)
                {
                    currentGrid[x, y, 0] = -1;
                }
            }
        }
    }

    //Sumbstract grid two from grid on)
    int[,,] GridSubstract(int[,,] arra1, int[,,] arra2)
    {
        int[,,] grid = new int[width, height, 1];
        for (int x = 0; x < width; x++)
```

```
        {
            for (int y = 0; y < height; y++)
            {
                if (currentGrid[x,y,0] != -1)
                {
                    grid[x, y, 0] = arra1[x, y, 0] - arra2[x, y, 0];
                }


            }
        }
        return grid;
    }

    //sets adjusted grid cell value to -1; can be called publicly
    public void SetVoxelValue(int[] value)
    {
        antiGrid[value[0], value[1], value[2]] = -1;
    }

    //Creates Object Pool / Queue
    void CreateObjectPool()
    {
        pool = new GameObject("WaterPool");

        objectPool = new Queue<GameObject>();

        int size = maxParticleCount;
        for (int i = 0; i < size; i++)
        {
            GameObject obj = Instantiate(waterParticle, pool.transform);
            obj.SetActive(false);
            objectPool.Enqueue(obj);
        }
    }

    //Deques object from object pool
    GameObject SpawnFromPool(Vector3 position, Quaternion rotation)
    {
        GameObject objectToSpawn = objectPool.Dequeue();
        objectToSpawn.SetActive(true);
        objectToSpawn.transform.position = position;
        objectToSpawn.transform.rotation = rotation;

        //objectPool.Enqueue(objectToSpawn);

        return objectToSpawn;
    }
    #endregion

    #region MeshGenerator

```

```
    Mesh mesh;

    Vector3[] vertices;
    int[] triangles;
    Vector2[] uvs;

    //Create initial vertice matrix
    void initializeVertices()
    {
        mesh = new Mesh();
        GetComponent<MeshFilter>().mesh = mesh;

        vertices = new Vector3[(width + 1) * (height + 1)];
        uvs = new Vector2[vertices.Length];

        for (int i = 0, y = 0; y <= height; y++)
        {
            for (int x = 0; x <= width; x++)
            {

                vertices[i] = new Vector3(
                                origin[0] + (-width / 2 + x) * particleScale - particleScale * 0.5f, origin[1] + 0.1f,
                                origin[2] + (-height / 2 + y) * particleScale - particleScale * 0.5f)
                                - gameObject.transform.position;

                uvs[i] = new Vector2(vertices[i].x, vertices[i].z);

                i++;
            }
        }
    }


    //Generate triangles using voxelMatrix and vertices
    public void CreateShape(int[,,] voxelGrid)
    {

        int[,,] grid = voxelGrid;

        triangles = new int[width * height * 6];

        int vert = 0;
        int tris = 0;

        for (int y = 0; y < height; y++)
        {
            for (int x = 0; x < width; x++)
            {

                if (grid[x, y, 0] == 1)
```

```
                {
                    triangles[tris + 0] = vert + 0;
                    triangles[tris + 1] = vert + width + 1;
                    triangles[tris + 2] = vert + 1;
                    triangles[tris + 3] = vert + 1;
                    triangles[tris + 4] = vert + width + 1;
                    triangles[tris + 5] = vert + width + 2;
                }

                vert++;
                tris += 6;
            }

            vert++;

        }

    }

    //Renders mesh
    public void UpdateMesh()
    {
        mesh.Clear();

        mesh.vertices = vertices;
        mesh.triangles = triangles;
        mesh.uv = uvs;
        mesh.RecalculateNormals();

        GameObject[] list = GameObject.FindGameObjectsWithTag("waterParticle");

        foreach (GameObject obj in list)
        {
            obj.SetActive(false);
            objectPool.Enqueue(obj);
        }
    }

    #endregion

    void DestroyObjectPool()
    {
        foreach (var pool in objectPool)
        {
            Destroy(pool);
        }
    }

    void ResetVariables()
    {
        nextVolume = 0;
        count = 0;
```

```
        volume = 0;
        spillArea = 0;
        alternate = false;
        mouseClicked = false;
        startTime = Time.time;
        partcileCountCheck = false;

        if (objectPool != null)
        {
            DestroyObjectPool();
        }
    }


}
```