Department of Civil Engineering

BIM-based construction simulation modelling

Carlos Arturo Osorio-Sandoval

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

March 2021

Declaration

I hereby declare that this thesis is the result of my own research and that the work has not been submitted for any other degree or professional qualification to any other institution other than the University of Nottingham for the degree of Doctor of Philosophy. I confirm that the work submitted is my own, except where commonly accepted ideas were adopted or where specific reference has been made to the work of others, to whom appropriate credit has been given.

Abstract

Construction simulation has been widely used in academia for research purposes and its usefulness as a planning and decision-making support tool has been proven. However, it has been neglected by the industry for various reasons, including the amount of data, skills, effort and time required to develop complex simulation models, the difficulty of model reuse, and the abstract and confusing way in which simulation results are usually presented.

This thesis proposes a simulation modelling approach that leverages existing simulation modelling paradigms used in the context of construction engineering and management, namely, discrete-event simulation, distributed simulation, hierarchical control structures and parametric modelling. The proposed simulation modelling approach enables an accurate representation of resource allocation and task interdependencies constraints while enabling model reuse to streamline the process of developing complex simulation models. Moreover, the proposed simulation approach provides a mechanism to enhance the visualisation of simulation results by generating simulation-based animations, which can be used for different visualisation purposes. The thesis discusses how the proposed simulation approach could tackle some o f the barriers to adopting simulation in the industry.

Subsequently, the thesis presents a framework for the semi-automatic generation of a construction simulation model and animations of its results from a building information model (BIM). The development of the proposed conceptual framework was based on the proposed simulation modelling approach. The framework is composed of five main modules: (1) the environment module, preloaded with a library of generic simulation models of different construction activities, (2) the user input, which includes the facility to import existing BIM models, (3) the pre-processing module, which automatically generates a BIM-based simulation model, (4) the simulation module, in which users can experiment with the model, and (5) the visualisation module, which produces reports and simulation-based animations to support planning and decision-making.

The proposed conceptual framework and its components were tested by designing a game engine-based application implemented in the Unity game engine. The features of the selected game engine were exploited to achieve the intended functionality of the framework. The feasibility of the framework was assessed through a case study based on a typical masonry construction problem.

Results of implementing the framework reveal that BIM-based simulations can reduce the skills, effort and time required to develop simulation models, and enable model reuse. The integration of simulation-based animations provides a model verification and validation mechanism, and a means to communicate model results to stakeholders that are unfamiliar with simulation.

Keywords: building information modelling, discrete-event simulation, construction simulation, simulation-based animation, game engine

Acknowledgement

Throughout the realisation of this project I have received a great deal of support and assistance. I would like to express my profound gratitude to my primary supervisor, Dr Walid Tizani, whose continual guidance and invaluable support was above and beyond the call of duty. I would also like to thank Dr Jelena Ninić and Dr Estacio Pereira for their continuous support, advice and motivation. My appreciation also extends to Dr Christian Koch for his tremendous support and for believing in me from the beginning. I would also like to thank Professor Haijiang Li and Dr Tao Liu for their insightful comments and valuable feedback during the defence of this thesis.

I wish to gratefully acknowledge the financial support provided by the Consejo Nacional de Ciencia y Tecnología (Conacyt), Mexico, during the entirety of my PhD studies, and by the Secretaría de Educación Pública, Mexico, during the 2018-2019 period.

I am also grateful to all members of the Centre for Structural Engineering and Informatics research group for their feedback at seminars, and above all, for their encouragement and friendship. Many thanks to Alison Johnson, Dr Elizabeth Davey and to all my friends from the Graduate School for their kind advise and motivation. My special thanks to Prof Jan Bradley and to all my friends from the Pastoral Care system. Great thanks to everyone with whom I shared my academic journey at the University of Nottingham. I am grateful for their friendship and for providing a happy distraction to rest my mind from my research.

Lastly, I want to heartily thank my family. I am grateful to my parents, Aurora and Ramón, for their wise advice that have helped me navigate through life. I would not be where I am without them. To my siblings, Alex and Bety, for their love and support that has been a source of motivation and inspiration. To my extended family for their encouragement and for believing in me. To my mother in law, Ana Lilia, and her extended family for their amazing support.

I am most grateful to my beloved wife and life partner, Ana Irene, for being by my side through laughter and tears. This thesis would not exist without her. She deserves more than I could ever give.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

The purpose of this chapter is to introduce the research by providing an overview of its background and a problem statement, which highlights the motivation to conduct it. The chapter also states the research hypothesis and research questions. Furthermore, it presents the highlights of the study and its contribution to knowledge. Finally, the chapter outlines the content of the thesis.

1.1 Background

Construction simulation is the method of modelling construction systems to understand their behaviour and experimenting with the model on a computer (AbouRizk et al. 2011). Simulation has become the method of choice for comparative research as it provides a means to test what-if scenarios in a timely manner (Hamdan et al. 2015; Ben-Alon and Sacks 2017). Traditionally, the prevalent approach for simulating construction systems has been discrete-event simulation (DES) (AbouRizk 2010). DES refers to modelling a system as it evolves by a representation in which state variables change instantaneously at separate points in time at which an event occurs (Law and Kelton 2000).

DES systems have been successfully implemented at the operational and project levels independently in a wide range of applications in research (D.-E. Lee et al. 2010). Some examples of these applications are estimating the impact of diverse variables on the duration of construction projects and operations (Ebrahimy et al. 2011; Werner and AbouRizk 2015; Osorio-Sandoval et al. 2016; Seo et al. 2016; Larsson and Rudberg 2019), studying and improving safety performance of construction projects (Pereira et al. 2018; Baniassadi et al. 2018), supporting decision-making and resource management (Zayed and Halpin 2001; C. Zhang et al. 2008; Feng et al. 2018), and studying and improving construction operations (Rahm et al. 2016), among others.

Simulation studies typically consist of four elements, namely, the input data, a process model, a simulation tool and the results (Puri and Martinez 2013). Each of these elements presents challenges that have hindered the prevalence of simulation modelling in the construction industry. Over the last four decades, numerous researchers in the field have addressed them both separately and combined. Their efforts have accomplished advancements in the modelling tools and proved that simulation has the potential to enhance decision-making and support planning. However, there are still barriers that prevent its widespread adoption by the industry, such as the amount of data, skills, effort and time required to develop complex simulation models, the difficulty of model reuse, and the abstract and confusing way in which simulation results are usually presented (AbouRizk 2010; S. Lee et al. 2013; Leite et al. 2016; Abdelmegid et al. 2020).

On the other hand, building information modelling (BIM) is one of the most promising recent developments in the architecture, engineering, and construction industry. With BIM technology, an accurate virtual model of a building is digitally constructed (Eastman et al. 2011). Such a model contains precise geometry and relevant data needed to support the design, procurement, fabrication and construction activities required to realise the building (Azhar 2011). In contrast with construction simulation, BIM technologies have been quickly adopted by industry practitioners, as a wide variety of stakeholders are demanding BIM and changing contract terms to enable it (Eastman et al. 2011). In 2018, 74% of the large architecture, engineering and construction (AEC) companies in the UK were aware of and using BIM, and it was projected that this number would increase to 90% within three to five years (*The National BIM Report 2018* 2019).

In recent years, researchers have attempted to integrate BIM and construction simulation techniques in an effort to overcome the barriers to adopting simulation in the industry (König et al. 2012; W.-C. Wang et al. 2014; W. Lu and Olofsson 2014; H. Liu et al. 2015; S. Chang et al. 2015; C. Wu et al. 2016; Barkokebas et al. 2017). However, most attempts have relied on using BIM-generated quantity take-offs as input for the simulation models, which on its own is not sufficient for detailed construction simulations. Constraints related to resource allocation and task interdependencies should also be considered to represent accurately the nature of construction. Moreover, a methodology to achieve this integration while exploiting the benefits of both technologies in a simple, expandable and customisable way is still needed. Among these benefits, 3D visualisation of simulation results should not be overlooked.

This study presents a framework that integrates BIM and construction simulation. The simulation approach of the proposed framework leverages various modelling paradigms to account for their individual shortcomings, including distributed simulation, hierarchical control structures and parametric modelling. The proposed framework semi-automatically generates a BIM-based construction simulation model that considers resource allocation and task interdependencies constraints. Furthermore, it produces simulation-based animations, which can be used for model verification and validation, or for other visualisation purposes. The framework was implemented within a game engine. The features of the selected game engine were leveraged to achieve the sough functionality of the proposed framework. The feasibility of the proposed framework is assessed through a case study on a typical masonry construction problem.

1.2 Problem statement

In the context of this research, construction simulation refers to modelling the construction process of a facility at the operational level. These models are used to represent the sequence of tasks or activities of a construction operation or project, their key performance indicators, such as duration and cost, and the utilisation of different resources, including materials, equipment, labour and space during the execution of such tasks.

Even though construction simulation is a valuable tool for planning and decision-making support, it has been neglected by the industry for many reasons. These include the amount of data, skills, effort and time required to develop complex simulation models, the difficulty of model reuse given the uniqueness of every project in the AEC industry, and the abstract and confusing way in which simulation results are usually presented. As a result, the benefits that simulation could bring to AEC companies and to the construction industry in general, are not exploited.

In an effort to overcome the above mentioned issues, researchers have proposed to use BIM data as input for simulation models with limited documented success (König et al. 2012; W.-C. Wang et al. 2014; W. Lu and Olofsson 2014; H. Liu et al. 2015; S. Chang et al. 2015; C. Wu et al. 2016). The integration of BIM and simulation has many potential uses both in industry and in research. Such an integration must consider resource allocation and task interdependencies constraints, which are fundamental to construction projects. Moreover, model reuse and visualisation of simulation results must not be overlooked by BIM and simulation integration approaches.

Therefore, in this research, a simulation approach that considers the typical constraints of construction projects is proposed first. The proposed approach aims to enable model reuse while levering BIM models. In this approach, BIM models provide parameters to generic simulation models of each construction activity required to build each element in the BIM model. Second, a framework based on such an approach is developed and implemented within a game engine, with the objective of taking advantage of the game engine's capabilities to produce simulation-based animations for visualisation purposes, including verification and validation of the resulting simulation model. Finally, the implemented framework is tested through a case study to illustrate and assess its usefulness as a tool to support planning.

1.3 Aim and objectives

The primary aim of this research is to investigate how to use a BIM model to facilitate the development of a construction simulation model that takes into account constraints related to resource allocation and task interdependencies. The following objectives have been identified to achieve the aim of this study:

- **Objective 1:** To review existing simulation approaches used in the construction context and existing techniques applied to visualise simulation results.
- **Objective 2:** To design a framework to generate a simulation model from a BIM model and show its results in the form of animations.

- **Objective 3:** To develop a game engine-based application based on the proposed conceptual framework that enables achieving its sought functionality.

- **Objective 4:** To assess the usefulness of the implemented framework through a case study.

1.4 Research hypothesis

Based on the aim and objectives outlined in the previous section, this research has adopted the following hypothesis:

A BIM-based construction simulation modelling approach can allow a semi-automatic development of complex construction simulation models by enabling model reuse and providing a means to visualise simulation results.

1.5 Research questions

The hypothesis was decomposed in the following research questions:

Q1 How is simulation used in the construction context?

Q2 What are the current trends in the field of construction simulation?

Q3 How are constraints related to resource allocation and task interdependencies represented in construction simulation models?

Q4 How can model reuse be leveraged in the simulation domain?

Q5 How can the visualisation of simulation results be enhanced?

Q6 How can the features of a game engine be leveraged to achieve the sought functionality of the proposed framework?

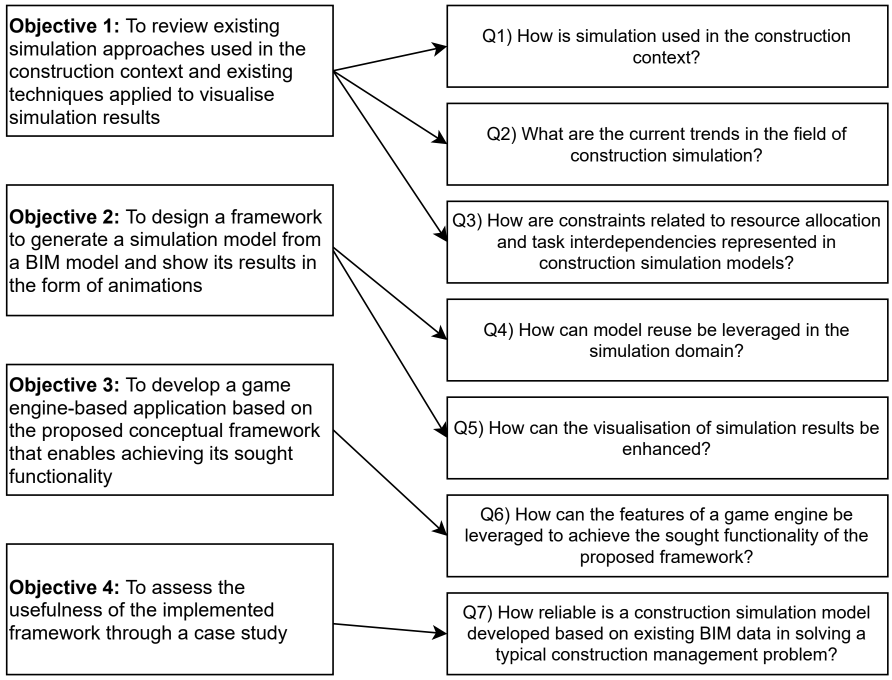


Figure 1.1: Research objectives mapped to research questions

Q7 How reliable is a construction simulation model developed based on existing BIM data in solving a typical construction management problem?

Figure 1.1 shows how the research questions are mapped to the research objectives previously outlined.

1.6 Research contribution

The contribution of this research mainly lies in three aspects:

1. The proposed integrated simulation modelling approach that leverages various modelling paradigms and accounts for their individual shortcomings. This approach enables model reuse and considers resource allocation and task interdependencies constraints.

2. The development of a framework, based on the proposed simulation approach, that is capable of semi-automatically generating a BIM-based con-

struction simulation model.

3. The demonstration of the usefulness of the implemented framework as a planning tool developed within a game engine.

Such contributions can be summarised in the following highlights:

- This research presents a framework that semi-automatically generates a BIM-based simulation model.
- The framework produces animations to communicate simulation results.
- The framework leverages BIM, construction simulation and game engines.
- The framework reduces the skills, effort and time required to develop simulation models.

1.7 Thesis layout

The remaining of this thesis is organised as follows:

Chapter 2: Literature review

This chapter reviews the existing state of knowledge regarding the field of simulation in construction. It explores the latest research concerning the use of BIM models to facilitate the development of construction simulation models. It also reviews approaches for producing simulation-based animations for visualisation purposes. The chapter also identifies the gaps in knowledge and the work required to address them, to which this thesis contributes.

Chapter 3: Research methodology

This chapter outlines the methodology adopted in this research work.

Chapter 4: A framework for the integration of construction simulation and building information modelling

This chapter presents a framework for the integration of construction simulation and BIM. The framework semi-automatically generates a BIM-based construction simulation model that considers resource allocation and task interdependencies constraints. Furthermore, it produces simulation-based animations, which can be used for model verification and validation, or for other visualisation purposes. The chapter provides an overview of the proposed framework first. The system requirement analysis to design the proposed framework are also outlined. Finally, the chapter describes the framework's five main modules and their components, as well as the required user input.

Chapter 5: Implementation of the proposed framework within a game engine

This chapter describes the implementation of the developed framework for the integration of construction simulation and BIM within a game engine. It discusses the game engine capabilities leveraged in the proposed framework, and details how they were used to model its components to deliver the desired functionality.

Chapter 6: An illustrative case study to assess the feasibility of the implemented framework

This chapter demonstrates the employment of the implemented framework through a case study that seeks to assess the usefulness of the framework as a planning tool. The case study also seeks to illustrate the development of the required components of the framework and guides users on how to design customised environments based on the proposed framework. The obtained results from the case study are also analysed and discussed to evaluate the framework's capabilities in regards to scenario development.

Chapter 7: Conclusions and recommendations

This chapter summarises the contribution and achievements of this research. It discusses how the objectives of the researchers were achieved and provides conclusions relevant to those objectives. Additionally, the chapter discusses the limitation of the work, and provides recommendations for future research.

Chapter 2

Literature review

This chapter explores the latest research concerning simulation in construction. It provides an overview of the relevant principles and theories from the field of simulation, and their adoption and evolution within the construction domain. The chapter also addresses the current trends within the field of simulation, with a particular focus on the use of BIM models to facilitate the development of simulation models. Furthermore, it reviews current approaches for producing animations to enhance simulation visualisation. Subsequently, this chapter identifies the gaps in knowledge and the work required to address them, to which this thesis contributes.

2.1 Simulation in construction

Simulation can be defined as the process of numerically evaluating a model with a computer, and gathering data to estimate the characteristics of such a model. In this context, a model is a set of assumptions about how a physical or abstract system of interest works. Simulation is needed when the relationships that compose a model are too complex to allow an analytic solution (Law and Kelton 2000). Such is the case of the construction process of facilities, which becomes more difficult to manage using other techniques as the projects increase in size or complexity (AbouRizk 2010). In a real-world construction project, activities are often subject to considerable uncertainty due to many different factors. They may take more or less time than originally estimated, resources may become unavailable, materials may arrive behind schedule, workers may be absent, and weather conditions may cause severe delays (Li et al. 2012), to name but a few examples.

Therefore, construction simulation is the method of developing and experimenting with computer-based representations, or models, of construction systems to understand their underlying behaviour and, subsequently, develop better project plans, optimise resource usage, minimise costs or project duration, or improve overall construction project management (AbouRizk 2010). Equally, simulation is also useful to identify critical elements of a system, such as weak or pinch points, and to examine the effects of changes to a system that would be impractical to implement in the real system (Long 2010). Thus, construction simulation has become the method of choice for comparative research as it provides a means to test what-if scenarios in a timely manner (Hamdan et al. 2015; Ben-Alon and Sacks 2017).

While there are other simulation applications in the construction domain, such as indoor ventilation, building energy performance, lighting and daylighting or indoor acoustic simulation (H. Wang and Zhai 2016), in the remainder of this document, the term simulation is used to describe the simulation of construction processes at the operational level. These models are used to represent the sequence of tasks or activities of a construction operation or project, their key performance indicators, such as duration and cost, and the utilisation of different resources, including materials, equipment, labour and space during the execution of such tasks.

Simulation models are usually classified along the following four different dimensions (Law and Kelton 2000):

1. *Static vs. dynamic simulation models*: Static simulation models are used to represent systems in which time does not play a role, while dynamic simulation models represent systems as they evolve over time.

2. *Deterministic vs. stochastic simulation models*: Stochastic simulation models contain probabilistic components and produce a different outcome each time they are run whereas deterministic models produce the same output for a given input.

3. *Continuous vs. discrete-event simulation models*: In discrete-event simulation models the state variables change at separated (discrete) points in time in which events occur, where an event is an instantaneous occurrence that changes the state of the system. Discrete-event simulation models are generally dynamic and stochastic in nature. In contrast, in continuous models the state variables change continuously with respect to time.

4. Local vs. distributed simulation models: Local simulations represent single applications in one computer, while distributed simulations consist of several simulation models that communicate through a computer network (Taghaddos 2010). A very different way to distribute a simulation, adopted in this research, is to decompose the model into several submodels (Law and Kelton 2000) that do specific functions while maintaining interoperability between them (AbouRizk 2010).

Generally, in construction simulation models the duration and cost of construction activities are considered as stochastic variables represented by probability distributions obtained from historical data sets, or assumptions on expected, optimistic and pessimistic values, which yield triangular distributions. The outputs of interest are analysed as the model evolves in time, and when, depending on the granularity of the model, tasks or activities occur. Therefore, the prevalent approach for simulating construction systems has traditionally been discrete-event simulation (DES) (AbouRizk 2010).

2.1.1 Discrete-event simulation in construction

In the construction domain, DES has been successfully implemented in a wide range of applications in research both at the operational and project levels.

DES has been used to support decision-making in regards to resource utilisation by offering the possibility to analyse the effect of different resource management strategies on one or more performance indicators of interest. For example, DES was applied to evaluate how different resource combinations impacted the production time and cost of concrete based on mix design and transportation distance from the plant on concrete batching operations (Zayed and Halpin 2001), and on the flow of operations and cost effectiveness of concrete paving operations (Hassan and Gruber 2008). Furthermore, DES was used to optimise the resource combination for cost and time in bridge deck rehabilitation projects (C. Zhang et al. 2008) and in asphalt concrete pavement operations (Younes et al. 2020), and to optimise the resource combination and material consumption for environment, cost and time in hotel building projects (Feng et al. 2018). By comparing different scenarios, managers can make an informed decision at the planning stage of their projects, thus, reducing uncertainty and risk.

DES has also been used to provide decision support on supply chain strategies in the construction domain. This approach can effectively model the material demand of different activities comprising construction projects. For example, DES was applied to show the impact of supply chain issues on the duration of tunnelling projects (Ebrahimy et al. 2011), and to calculate construction objectives under different supply chain strategies and construction configurations in pre-cast concrete projects (S. Chen et al. 2019). This approach underlines the importance of considering external aspects as integral parts of construction projects that can also affect their performance.

DES is a useful methodology to study the individual impact of variables that affect construction productivity but that cannot be isolated or controlled in real-life projects. For example, DES has been used to analyse the effect of equipment breakdown on productivity performance of tunnelling projects (Werner and AbouRizk 2015; Rahm et al. 2016), to estimate quantitatively the effect of labour absenteeism on the duration of construction activities in housing projects (Osorio-Sandoval et al. 2016), to quantitatively evaluate the impact of workers' muscle fatigue on time and cost of construction operations (Seo et al. 2016), and to study the impact of temperature, precipitation and wind speed on in situ wall operations (Larsson and Rudberg 2019). This technique enables further analysis of the mitigation of delays and cost overruns that these variables may produce, and the possibility of designing optimised operations considering their effect. For example, Conrads et al. (2017) combined DES with other simulation approaches to evaluate different maintenance strategies for equipment used in tunnelling projects to mitigate the bottlenecks identified by Rahm et al. (2016).

DES has been used in construction safety applications to assist managers with proactive development of strategies to improve safety. For example, Pereira et al. (2018) combined DES and continuous simulation to predict how resource allocation and safety policies affected safety performance on construction projects. Baniassadi et al. (2018) used DES to compare scenarios of complex and hazardous construction operations with different levels of safety and productivity to help managers planning concurrently for the improvement of these two variables. Other applications of DES that involve comparing different scenarios to support decision-making are, to mention but a few examples, the evaluation of the impact of lean construction principles on the performance of reinforcement operations (Bajjou and Chafi 2020), and modelling the inspection processes and the behaviour of technicians, traffic and testing devises to support planning of concrete bridge deck inspections (Abdelkhalek and Zayed 2020).

2.1.2 Simulation tools used in construction

One of the first DES-based simulation languages specifically designed to model construction operations is the Cyclic Operations Network (CYCLONE) technique (Halpin 1977). In CYCLONE models, construction operations are represented using cyclic Petri networks, which are a natural means to describe repetitive processes and enable both the analysis and estimation of construction productivity. An identified limitation of CYCLONE models is their inability to explicitly represent resources (AbouRizk 2010).

Several CYCLONE-based simulation tools have been developed to enhance and extend the capabilities of the CYCLONE methodology. Examples include INSIGHT (Kalk 1980), UM-CYCLONE (Ioannou 1989) and MicroCYCLONE (Halpin 1990), the latter with examples of successful implementation documented in the literature more than a decade after its development (Zayed and Halpin 2001; C. Zhang et al. 2008). Furthermore, tools like RESQUE (D. Y. Chang and Hoque 1989) enhance the CYCLONE concepts and functionalities by implementing an object-oriented approach that allows the distinction of resources and defining their interactions without complicating the model. Other examples of CYCLONE enhancements that implement such an approach are COOPS (L.-Y. Liu 1991), DISCO (R.-Y.

Huang and Halpin 1994) and CIPROS (Odeh 1992).

Among the limitations of these tools, the extensive technical training and hands-on experience that are required to use them properly, and the limitation on modelling and simulation capabilities stand out. Efforts to overcome these issues include ABC (Shi 1999), which intended to make the simulation process as simple as the critical path method (CPM) by using a single element to model construction operations. Regardless, these efforts have not been embraced by industry practitioners.

Two of the most prevalent languages for special purpose construction simulation are Stroboscope (an acronym for STate- and ResOurce-Based Simulation of COnstruction ProcEsses) (Martinez 1996) and Simphony (Hajjar and AbouRizk 1999). Stroboscope models consist of a series of programming statements that define a network of interconnected modelling elements, give the elements their unique behaviour, and control the simulation (Martinez 1996). Recent examples of implementations of Stroboscope and its simplified version EZStrobe (Martinez 2001) can be found documented in the literature (Hassan and Gruber 2008; Puri and Martinez 2013; Seo et al. 2016; Younes et al. 2020). Simphony simplifies and standardises the development and utilisation of construction special purpose simulation tools by allowing for the construction of hierarchical and modular simulation models using extendible templates (Hajjar and AbouRizk 1999). Several examples of Simphony implementation for multiple construction applications are documented in the literature (Ebrahimy et al. 2011; Werner and AbouRizk 2015; Osorio-Sandoval et al. 2016; Pereira et al. 2018). One of the key advantages of both Stroboscope and Simphony lies on the possibility of writing custom programming code to manipulate the model and its components for more accurate and flexible modelling (AbouRizk 2010).

Simulation software packages not specifically design for the construction domain have also been used by researchers in this field. Examples include AnyLogic (Rahm et al. 2016; Conrads et al. 2017; Baniassadi et al. 2018; P. Zhang et al. 2019; Abdelkhalek and Zayed 2020), Simio (Song and Eldin 2012; W. Lu and Olofsson 2014; Feng et al. 2018; S. Chen et al. 2019), and Arena (Nikakhtar et al. 2011; Bajjou and Chafi 2020).

2.1.3 Limitations of construction simulation

Even though construction simulation has been proven to be a valuable tool for planning and decision-making support, it has been neglected by the industry for many reasons. A few comprehensive studies aiming at identifying the key challenges and barriers that have hindered the adoption of simulation techniques in practice have been carried out in the past decade (S. Lee et al. 2013; Leite et al. 2016; Abdelmegid et al. 2020).

The most frequent barriers reported in the literature are related to the limitations of the construction simulation tools and methods, which are often unable to represent the reality of construction systems (Abdelmegid et al. 2020). Moreover, the typical modelling approaches adopted in construction simulation, which are often borrowed from other industries like manufacturing, lack the generality to easily capture the modelling needs or complex decision-making mechanisms of construction (Golzarpoor et al. 2017), such as resource allocation and task interdependencies constraints. In turn, this leads modellers to come up with complicated workarounds, thus, increasing, rather than reducing, the amount of effort, skills and time required to develop useful simulation models. While there have been several research attempts to provide more powerful simulation tools, these have not been as prevalent in practice as those that offer modelling flexibility (AbouRizk 2010).

Another of the identified barriers to adopting simulation modelling in the construction industry is the fact that developing the process model takes a lot of time, a high level of effort, and requires the acquisition of specialised skills (W.-C. Wang et al. 2014; Abdelmegid et al. 2020). However, attempts to simplify model development have not been enough to enable the adoption of simulation in the industry (AbouRizk 2010). In this regard, the reuse of simulation models is appealing, based on the intuitive argument that it should overcome these issues (Robinson et al. 2004) while leading to higher quality simulation studies (Kasputis and Ng 2000).

Model reuse ranges from using small portions of code to full model reuse. While reuse at the higher end of the spectrum is problematic, reuse at the lower level is more manageable (Robinson et al. 2004). Model reuse is an identified benefit of the distributed simulation approach, described below. It is also one of the justifications of the High Level Architecture (HLA), a technical architecture developed to facilitate the reuse and interoperation of simulation systems and assets (Dahmann et al. 1998; IEEE Computer Society 2010).

However, model reuse in construction simulation still faces some fundamental issues. The nature of construction presents the challenge of modelling the dynamic allocation, transit, and matching of an assortment of resources at specific activity locations subject to various constraints (M. Lu and Wong 2007). These constraints are typically project-specific and vary across different projects, which generally makes model reuse not possible (AbouRizk et al. 2011; König et al. 2012). While the distributed simulation approach may be applied for model reuse by adopting a component-based development approach, this alone is not sufficient to overcome this issue, and re-formalisation of federates (submodels) would still be required when changes in the design model or in the project's conditions occur. In other words, the same model for a construction activity may not be equally valid for two projects with different constraints unless they are re-formalised accordingly.

Another identified barrier to adopting simulation modelling in the construction industry is the amount and nature of the input data needed to build simulation models (W. Lu and Olofsson 2014; Krantz et al. 2015; Abdelmegid et al. 2020). Although several tools are available for construction simulation, they all require proper input data to produce accurate results (Puri and Martinez 2013). However, preparing the input data for a simulation model is a time-consuming and error-prone process (I.-C. Wu et al. 2010; König et al. 2012). Moreover, there is also a significant amount of manual work needed to specify and maintain the interdependencies between activities and resources (W. Lu and Olofsson 2014). In recent years, researchers have turned to the integration of construction simulation with building information modelling (BIM), which has shown that it has the potential to overcome these issues. These research efforts are further described in Section 2.4.

Finally, another identified barrier to adopting simulation modelling in the construction industry that is related to the results from simulation models is that they are usually abstract and confusing (C. Wu et al. 2016). Because of that, decision-makers often misinterpret or fail to apply them. In turn, decision-makers prefer to delegate simulation to specialists and separate themselves from the process (Al-Hussein et al. 2006). All of this leads to a lack of trust by industry practitioners toward their effectiveness (Leite et al. 2016).

The remainder of this chapter focuses on the distributed simulation approach as adopted in this research, on the use of building information modelling (BIM) to facilitate the development of simulation models, and on current approaches for producing animations to enhance simulation visualisation.

2.2 Distributed simulation modelling and hierarchical control structures

Typically, distributed simulations consist of several simulation models that communicate through a computer network (Taghaddos 2010). In this research, a different way to distribute a simulation is adopted. In this modelling approach, several submodels (*federates*), such as process models of construction activities or models that represent the aforementioned factors that affect construction, are composed together (*federated*) to form an integrated model (*federation*) that represents the whole construction project. Figure 2.1 illustrates a typical architecture of the distributed simulation approach.

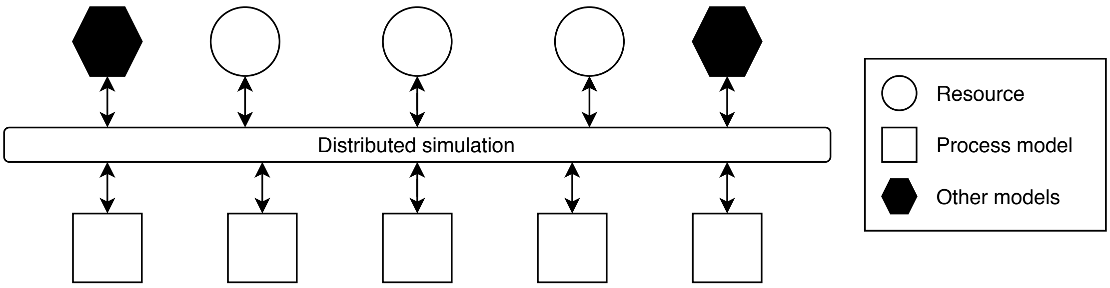


Figure 2.1: Architecture of a distributed simulation

A distributed simulation approach offers modellers enough flexibility to integrate different process models developed using different strategies. In the approach adopted in this research, each process model represents a different construction activity. Additionally, other elements such as resources or factors that affect construction, like equipment breakdown, weather, delays in procurement or labour absenteeism, can be modelled independently. This approach allows the capturing of the modelling needs of a typical construction project. Modellers are also able to scale down a process model that may not be of their interest, but that still plays a role in the simulated project. For example, resource utilisation of subcontracted activities may not be relevant in a simulation model developed to support resource allocation planning for a construction manager. However, their duration may affect the project's resource schedule for subsequent activities.

Distributed simulation allows modelling construction activities as independent federates. Each federate defines local interdependencies between the tasks that comprise the activity that they represent, as well as the resource utilisation during their execution. However, a construction project federation requires several more complex interdependencies to take into account the technological and strategic aspects of the project (König et al. 2012). The federation also requires a sophisticated mechanism to allocate shared resources. In this regard, Furian et al. (2015) introduced control units as part of the components of conceptual modelling. An explicit implementation of control units has the potential to meet both conditions.

Control units act in entity flow hierarchical control structures that add more flexibility in the abstraction and conceptual modelling process. Control units include a set of rules that determine the conditional behaviour of the model and remove the explicit use of queuing structures (Furian et al. 2015). The use of queuing structures is prevalent in construction simulation even though queuing structures are too rigid to model systems with complex behaviour (Golzarpoor et al. 2017). Figure 2.2 depicts the logic of a four-level control structure. As seen in the figure, this approach also contemplates modelling sequential tasks in a classical way (see tasks A and B). Besides determining the flow of entities in the model, control units also remove the need of classifying the role of entities (Furian et al. 2015). This allows modelled resources to change roles during the simulation between entities that serve and entities that are being served without modelling workarounds.

Implementing a hierarchical control structure to control entity flow in a distributed simulation provides a mechanism to allocate shared resources among tasks

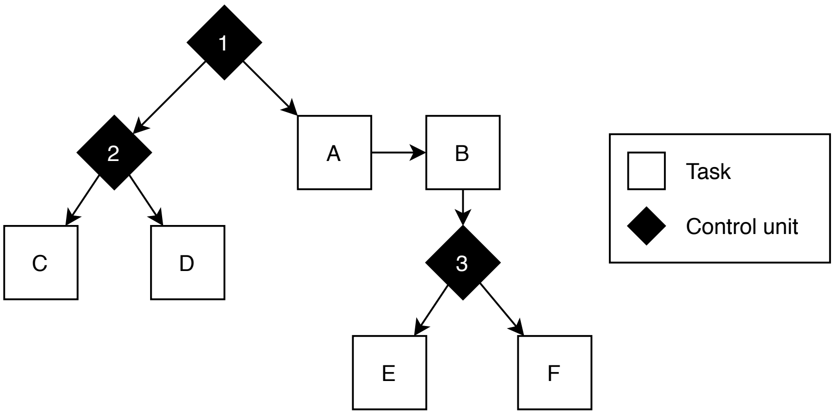


Figure 2.2: Logic of control structures

of different construction activity federates, and to comply with project-specific strategic and technological decision-making rules.

2.3 Parametric modelling

In the context of geometric modelling, parametric modelling is a modelling process with the ability to change the shape of model geometry as soon as the dimension value is modified (Fu 2018). As pointed out by G. Lee et al. (2006), parametric modelling is understood as the imposition of constrained relationships on the shape of objects rather than just as the definition of parameters. In this approach, new instances of an object can be created by assigning new values to a set of predefined parameters (G. Lee et al. 2006). Its key advantage is that the model geometry changes as soon as the parameters are modified, without the need to redraw the model whenever it needs a change (Fu 2018). The characteristics of this paradigm have made it attractive to many researchers, and both industry and academia have dedicated efforts to improving parametric modelling technology, as evidenced by the prevalence of building information modelling, for which 3D parametric modelling is central (G. Lee et al. 2006). However, in the construction simulation domain, this paradigm has not been exploited. Integrating BIM and construction simulation is a promising approach that could leverage this modelling paradigm.

2.4 Integration of building information modelling and construction simulation

Contrary to construction simulation, building information modelling (BIM) has been quickly adopted by industry practitioners, as a wide variety of stakeholders are demanding BIM and changing contract terms to enable it (Eastman et al. 2011). In 2018, 74% of the architecture, engineering and construction (AEC) companies in the UK were aware of and using BIM, and it was projected that this number would increase to 90% within three to five years (*The National BIM Report 2018* 2019). With BIM technology, an accurate virtual model of a building is digitally constructed (Eastman et al. 2011). Such a model contains precise geometry and relevant data needed to support the design, procurement, fabrication and construction activities required to realise the building (Azhar 2011). With this advancement in computer technologies, the geometric dimensions of building components can be added to process simulation models in order to obtain valuable insight into the details of construction activities that are difficult to represent (W.-C. Wang et al. 2014).

As previously discussed, developing and maintaining construction simulation models is a process that takes a significant amount of time, a high level of effort, and requires the acquisition of specialised skills (W.-C. Wang et al. 2014; Abdelmegid et al. 2020). Furthermore, simulation models require proper input data to produce accurate results (Puri and Martinez 2013), and, given the amount and nature of such data, preparing it is also a time-consuming and error-prone process (I.-C. Wu et al. 2010; König et al. 2012; W. Lu and Olofsson 2014; Krantz et al. 2015;

Abdelmegid et al. 2020). Moreover, dynamic changes to physical aspects of the simulated projects, such as walking distances and obstacles, need to be artificially pre-programmed in traditional simulation tools, since they do not enable the simulation of the physical environment (Ben-Alon and Sacks 2017). In an effort to overcome these issues, researchers have attempted to integrate BIM with construction simulation, leveraging BIM models in different ways.

BIM-generated quantity take-offs can be used to calculate the duration of activities in construction operation models (König et al. 2012; W.-C. Wang et al. 2014; W. Lu and Olofsson 2014; Barkokebas et al. 2017) and to determine the number of entities that must be processed in a model to complete an operation (Hamdan et al. 2015). BIM data can also be used as input for simulation models through databases (C. Wu et al. 2016). These approaches on their own, however, are not sufficient to generate detailed construction simulation models, since constraints related to resource allocation and task interdependencies cannot be determined using solely quantity take-offs.

Data extracted from BIM models can be enriched with additional information, such as topology, connections, supports, functionalities and work breakdown structure (WBS) information to guide entities through simulation models (H. Liu et al. 2015) and to use as simulation resource data (S. Chang et al. 2015; Jeong et al. 2016). This additional information could provide a means of considering relevant data that is not necessarily included in the design BIM model, such as formwork areas or temporary structures, in the simulation model. Furthermore, considering the topological relationships between different elements within a BIM model also provides the opportunity to automatically establish task interdependencies in the simulation model according to rules based on such relationships.

Another approach to integrate construction simulation models and BIM consists of assigning simulation models of construction activities to individual building components of a design model to interactively define complex simulation models (I.-C. Wu et al. 2010; König et al. 2012; Hamdan et al. 2015). This approach offers the additional benefit of enabling the reuse of simulation models across different projects, which is generally not possible in the context of construction simulation (AbouRizk et al. 2011). The following section addresses the reasons why simulation model reuse is appealing yet challenging.

2.5 Simulation model reuse

The reuse of simulation models is appealing, based on the intuitive argument that it should reduce the amount of time and effort required to build new models (Robinson et al. 2004) while leading to higher quality simulation studies (Kasputis and Ng 2000). Model reuse ranges from using small portions of code to full model reuse. While reuse at the higher end of the spectrum is problematic, reuse at the lower level is more manageable (Robinson et al. 2004).

Combining simulation models with BIM by assigning "modular" simulation models of activities to BIM elements to define a more complex simulation model that encapsulates the construction of the BIM elements in the BIM model is a promising approach to enable model reuse. For example, König et al. (2012) presented an approach to generate input data for construction simulations by using linked BIM data and two types of reusable templates: process pattern assignment templates, used to automatically assign process patterns to building elements, and interdependency templates, which reduce the effort to define individual interdependencies between processes. Equally, W. Lu and Olofsson (2014) proposed a framework in which BIM components are linked with construction recipes that link the product information with the process information to build it. Both approaches allow exploration of changes in the BIM model without manually checking and re-formalising the corresponding simulation models. However, these "modular" simulation models must be defined following an approach that allows model reuse since the conceptual modelling stage. To this end, in this research, two modelling paradigms are combined with BIM to enable simulation model reuse: distributed simulation and parametric modelling.

The distributed simulation paradigm allows existing models to be composed together (federated) to form simulations of large-scale systems (Anagnostou and Taylor 2017). In this approach, independently developed submodels (also called *federates*) can be reused to develop an integrated model (referred to as a *federation*) in a "grab and glue" fashion. In fact, model reuse is an identified benefit of this simulation approach. It is also one of the justifications of the High Level Architecture (HLA), a technical architecture developed to facilitate the reuse and interoperation of simulation systems and assets (Dahmann et al. 1998; IEEE Computer Society 2010).

On the other hand, in the context of geometric modelling, parametric modelling is a modelling process with the ability to change the shape of model geometry as soon as the dimension variable value is modified (Fu 2018). In this approach, new instances of an object can be created by assigning new values to a set of predefined parameters (G. Lee et al. 2006). Its key advantage is that the model geometry changes as soon as the parameters are modified, without the need to redraw the model whenever it needs a change (Fu 2018). The characteristics of this paradigm have made it attractive to many researchers, and both industry and academia have dedicated efforts to improving parametric modelling technology, as evidenced by the prevalence of BIM, for which 3D parametric modelling is central (G. Lee et al. 2006). However, in the construction simulation domain, this paradigm has not been fully exploited.

These paradigms can be combined together to enable simulation model reuse. Chapter 4 further details the simulation approach adopted in this research, which is based on these simulation paradigms and on the use of BIM data as source of project and product specific parameters that enable the automatic generation of simulation models with model reuse in mind.

2.6 Visualisation of simulation results

Typically, the outputs of simulation studies are presented in an abstract and confusing way (C. Wu et al. 2016), consisting of statistical tables and charts, which are seen as impractical by construction practitioners, particularly when the modelled system is complex (Abdelmegid et al. 2020). It is not uncommon for industry practitioners and decision-makers to view simulation modelling as a "black box" system that can only be understood by those who are familiarised with it. Coupled with the lack of simulation knowledge among practitioners, this issue leads decision-makers to misinterpret or fail to apply simulation results in their projects. At best, decision-makers prefer to delegate simulation to specialists and separate themselves from the process (Al-Hussein et al. 2006). All of this leads to a lack of confidence by construction practitioners toward the effectiveness of simulation modelling (Leite et al. 2016), which has also hindered its adoption in the construction industry (Abdelmegid et al. 2020).

Aligned with recent trends in the field of simulation, which consist of supporting efficient integration between simulation models and other tools and software (W. Lu and Olofsson 2014), researchers in the construction simulation domain have a special interest in enhancing visualisation (AbouRizk 2010). In this regard, simulation-based animations are viewed as appropriate tools to reduce the complexity in which simulation results are presented to stakeholders that unfamiliar with simulation.

2.6.1 Animations

In this context, an animation consists of displaying graphically the operational behaviour of a simulation model as it evolves through time (Sargent 2013). Compared to traditional 4D modelling, or 4D simulations, which only allow for visualisation of construction schedules, remaining on the level of visualisation and not planning (Jeong et al. 2016), simulation-based animations depict the construction processes more accurately. This is achieved by displaying how the resource interact with each other and how the construction site changes as the construction activities are carried out during the simulation.

The most prevalent approach to generate simulation-based animations is based on the post-processing visualisation concept (ElNimr and Mohamed 2011), which relies on generating "trace files" during the simulation as it runs. Each trace file contains all the information regarding the changes that the modelled system went through at the moment of its creation. A key requirement of trace files is that they must be time-stamped so that they can be ordered chronologically to track the changes in the system as they occurred during the simulation. Subsequently, the system uses the trace files to generate an animation that replicates what happened during the simulation after it finishes (Kamat and Martinez 2001; Kamat and Martinez 2003).

Simulation-based animations in the context of construction management have been used for diverse purposes. For example, Kamat and Martinez (2003) used 3D visualisations to verify and validate the DES model of an earthmoving operation with a complex control logic. In fact, animation is one of the techniques used for validating and verifying submodels and overall models frequently found in the literature of the simulation domain (Sargent 2013). Besides verification and validation, Al-Hussein et al. (2006) used 3D animations based on simulation models of tower crane operations to communicate the simulated operation and provide detailed information to decision makers. Rekapalli and Martinez (2010) also used 3D visualisations with the purpose of validating and verifying simulation models of earthmoving operations. Compared with previous research, they enabled user interaction with the model to generate scenarios of interest.

2.6.2 Using game engines

As new technologies become more available and cheaper, researchers in the field have turn to them to improve simulation-based animations. For example, ElNimr and Mohamed (2011) used a game engine to produce simulation-based animations of construction operations. The gaming industry is one of the leaders in utilizing and advancing visualisation technologies (ElNimr and Mohamed 2011). Gaming technologies, especially game engines, have drawn the attention of researchers in the construction simulation field. Game engines provide ways to achieve immersion in visualisation due to their compatibility with mixed reality technologies, ubiquity as they offer multi-platform deployment, high-quality graphics and network capabilities that allow for multiple simultaneous users, among other characteristics (Osorio-Sandoval, Tizani and Koch 2018). Such an approach enabled rapid prototyping and customisation of solutions with short decision-making time windows (ElNimr and Mohamed 2011).

Other examples of integration of construction simulation and new technologies include H.-M. Chen and P.-H. Huang (2013), who leveraged augmented reality to enhance visualisation of simulated operations, and both W.-C. Wang et al. (2014) and W. Lu and Olofsson (2014), who used traditional BIM 4D animations to display simulation results.

While there are many documented research attempts to improve the visualisation of simulation results through animations, none have leveraged both BIM data and game engines to accomplish this.

2.7 Gap in knowledge

It can be concluded from the literature review that the conventional implementation of construction simulation in the industry has not been exploited despite its proven efficacy. Numerous researchers have suggested that industry practitioners are discouraged from using simulation during the planning process of their projects due to the difficulties in model development and the inability to reuse hardily-developed models across multiple projects. Moreover, it is challenging for decision-makers to apply simulation results correctly due to the lack of confidence in them caused by the abstract and confusing way in which they are usually presented.

There is an essential need to streamline the model development process while enabling model reuse and enhancing the way in which the results are communicated to stakeholders that are unfamiliar with simulation. The distributed simulation approach coupled with hierarchical control structures can deal with interdependencies between tasks of different construction activity federates and provide a mechanism to allocate shared resources. This approach allows the capturing of the modelling needs of a typical construction project and its complex decision-making mechanism while offering modelling flexibility.

Integrating construction simulation with other tools and software is believed to be the future of this field of research (W. Lu and Olofsson 2014; Turner et al. 2016). In this regard, BIM has been identified as a potential tool to facilitate model development and to enable model reuse. This integration can also leverage the parametric modelling approach of BIM applied to simulation modelling. However, a methodology to achieve this integration while exploiting the benefits of both technologies is still missing from the literature. Among these benefits, simulation-based animations that enhance traditional 4D modelling must not be overlooked.

2.8 Summary

This chapter provided an overview of the latest research in the construction simulation field. It underlined the proven usefulness and efficacy of this method in construction engineering and management research and practice. The chapter revealed that industry practitioners tend to neglect construction simulation despite the many benefits that it offers to the AEC domain. Subsequently, the chapter focussed on research trends that have attempted to mitigate the challenges that hinder the adoption of simulation in the industry, particularly, the use of BIM to facilitate model development and simulation-based animations that enhance the visualisation of simulation results. Finally, the chapter identified the gaps in knowledge and revealed that there is a need to streamline the model development process while enabling model reuse and enhancement of results visualisation.

The next chapter explains the plan and design for the research project presented in this thesis.

Chapter 3

Research methodology

Based on the research findings from the literature review, a framework for the integration of construction simulation and building information modelling is proposed. The main purpose of such a framework is to streamline the process of developing complex construction simulation models while providing a mechanism to present simulation results to stakeholders that are unfamiliar with simulation.

This chapter outlines the adopted methodology for the development of the proposed conceptual framework and its components, which are further described in Chapter 4.

3.1 Framework objectives

The proposed framework aims to achieve the following objectives, drawn from the limitations outlined in the literature review:

1. Reduce the skills, effort and time required to develop construction simulation models.

2. Leverage BIM models to semi-automatically generate simulation models

based on BIM models.

3. Produce animations to communicate simulation results.

3.2 Design-science research methodology

This research adopted the design-science research methodology proposed by Hevner et al. 2004. The main result of this methodology is a purposeful artifact created to address an important problem, which has been outlined in Chapter 1. The artifact must be described effectively, enabling its implementation and application in an appropriate domain (Hevner et al. 2004).

In Chapter 2, the theories that support the artifact design are outlined. Chapter 2 aims to respond to the research questions **Q1** to **Q3**.

In Chapter 4, the proposed conceptual framework and its sought functionality are described. This conceptual framework is a representation of the artifact product of this research. Based on the literature review, specific requirements for the sought functionality of the proposed framework were set in Section 4.2. The components of the framework were designed iteratively. A hands-on approach was taken by testing and exploring several design options to ensure that the components responded to the outlined requirements. The five modules of the proposed framework, as well as their requirements, are described effectively. Chapter 4 further responds to the research questions **Q4** and **Q5**.

In Chapter 5, the proposed implementation of the conceptual framework in a game engine-based artifact is described. The chapter details the requirements of the components of the conceptual framework to implement them in a functional prototype. The functionalities of a game engine, namely, Unity were investigated and tested to justify their inclusion into the implementation of the system and into the conceptual framework. Chapter 5 aims to respond to the research question **Q6**.

In Chapter 6, an illustrative example of the implementation of the artifact in a typical problem of the construction management domain is presented in the form of a case study. The presented case study follows its own methodology described within the chapter. The purpose of the case study is to evaluate the employment of the artifact based on the proposed conceptual framework. Furthermore, the chapter also serves as a guide to enable users to develop, implement and apply the framework to other problems of the construction management domain. Chapter 6 aims to respond to the research question **Q7**.

3.3 Summary

This chapter described the methodology adopted in this research project. The next chapter provides a detailed description of the proposed conceptual framework for the integration of construction simulation and building information modelling, its design and components. The system requirement analysis is also provided in the next chapter.

Chapter 4

A framework for the integration of construction simulation and building information modelling

This chapter describes the design and development of a framework to integrate construction simulation and BIM. An overview of the proposed framework is presented first, followed by the system requirement analysis. Then, the environment module is discussed, which consists of three elements that enable the semi-automatic generation of BIM-based simulation models and simulation-based animations. The required user input to leverage the features of the proposed framework is described next, followed by the description of the preprocessing module, which develops the simulation model based on elements from the environment module and the user input. Then, the simulation module is discussed, which contains the required components to execute the simulation of the model. Finally, the visualisation module is described. This module uses simulation outputs to produces reports with relevant information to support decision-making during construction planning, as well as simulation-based animations for various visualisation purposes.

4.1 Framework design

Figure 4.1 illustrates the proposed conceptual framework for the integration of construction simulation and building information modelling and its components. As shown in the figure, the proposed framework is composed of five main modules: the environment, the user input, which includes the facility to import existing BIM models, a pre-processing stage, the simulation module and the visualisation module.

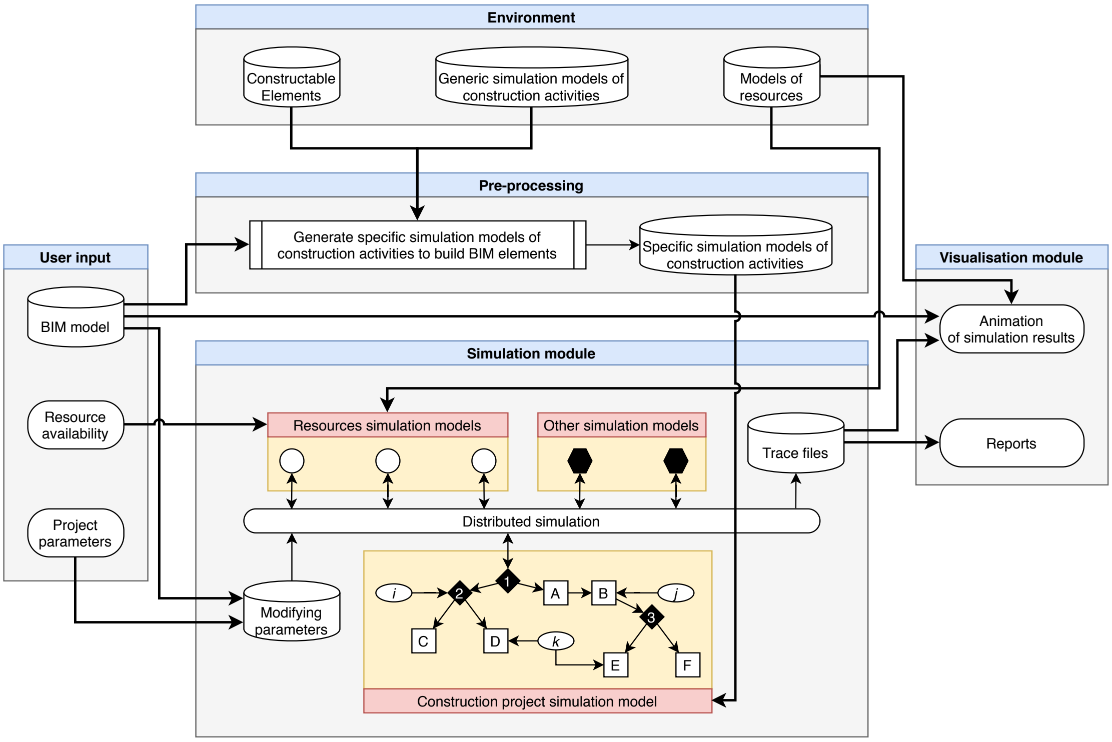


Figure 4.1: Framework overview

The user imports a BIM model into the environment and defines the project parameters related to resource constraints (e.g. number of workers and their skills, location of raw materials and available tools and equipment). The environment is preloaded with a library of generic simulation models of different construction activities. Based on the types and parameters of the elements of the input BIM model, a simulation model of the construction of the project is generated semi-automatically. The simulation of this model considers the resource constraints previously defined by the user. During the simulation, time-stamped trace files that indicate the state and location of resources in the construction site are generated.

In the visualisation module, the trace files are processed to produce reports to support planning and decision-making. Furthermore, the trace files, the BIM model and other 3D models that are preloaded into the environment to represent resources are used to create a simulation-based animation of the construction of the project. The resulting animation gets into the details of resource interaction and depicts the construction process at a high level-of-detail. The animation can be used for various visualisation purposes, including simulation model verification and validation, communicating results to stakeholders and as a decision-making and planning support tool. The animation can also be visualised using mixed-reality technology.

4.2 System requirement analysis

This section outlines the requirements for developing a system based on the conceptual framework proposed in the previous section. The modules of the proposed framework were designed based on these requirements.

4.2.1 Reuse of construction simulation models

As previously discussed, one of the limitations hindering the adoption of construction simulation in industry is the fact that reusing hardily developed simulation models across multiple projects is generally not possible due to project- and product-specific constraints that are prevalent in construction. Implementing a parametric simulation modelling approach aims to address this issue.

With a parametric simulation modelling approach, a simulation model can be scaled by parameters that represent some of the project-specific constraints that are prevalent in construction. Such parameters could modify directly one or more process models or the control units that determine their behaviour. For example, based on the height of a wall, it may be determined that a scaffolding system is required, thus, prompting the model to direct entity flow to a scaffolding installation federate. It is worth noting that the modifying parameters are not limited to geometric constraints. For example, resources may be allocated differently according to a parameter that establishes a limit in the number of workers on the site at the same time. In this regard, modifying parameters may be related to the product of the construction (product parameters), or to the constraints of the project (project parameters).

The proposed strategy also benefits from a parametric modelling approach by reducing the extensive manual modifications that simulation models are subject to when small changes occur in the project. This allows modellers to develop different scenarios to test construction alternatives quicker.

In order to fully take advantage of this approach, modellers should consider reuse at an early stage of model development. The modifying parameters and how they affect the fundamental models must be clearly documented. Reusable components should be modelled at a granularity that responds to the modifying parameters. Over time, as the library of fundamental models is built up, much more comprehensive representations of a construction project should be possible (Kasputis and Ng 2000).

While parametric simulation modelling is a promising approach to enable model reuse, there are two important challenges that must be considered. The first one is related to the complexity of parametric modelling. With large numbers of parameters, parametric modelling becomes a technical skill in its own right and communicating the model's behaviour represents a challenge (G. Lee et al. 2006). In this regard, conceptual modelling and proper documentation of the identified modifying parameters and their effect on the fundamental models are needed at all the levels of granularity. Equally, visualisation of the simulation results provides a means to communicate the behaviour of the model.

The second challenge has to do with the scalability of the random variables that simulation models take as input. As noted by Puri and Martinez (2013), conversion of data from one discretisation unit to another affects its statistical properties. In this regard, it is recommended to use the smallest possible discretisation unit and accumulate the data points together, rather than subdividing the data into smaller parts (Puri and Martinez 2013).

4.2.2 Simulation modelling flexibility

As discussed in Section 2.2, a distributed simulation approach offers modellers enough flexibility to integrate different process models developed using different strategies. Additionally, other elements such as resources or factors that affect construction, like equipment breakdown, weather, delays in procurement or labour absenteeism, can be modelled independently. In the distributed simulation approach adopted in this research, independent federates representing construction activities are composed together to form a federation that represents the construction process of a project. Such federates are parametric in nature, which enables model reuse as well as modelling flexibility.

4.2.3 Complex decision-making mechanism

The proposed system must address the complex decision-making mechanism of construction management. It also requires a mechanism to allocate shared resources.. An explicit implementation of control units Furian et al. (2015) in a distributed simulation approach has the potential to meet these conditions. Project- and product-specific parameters can be embedded into the logic of control units as parameter-based rules to adopt the parametric simulation modelling approach.

4.2.4 BIM-based simulation and the Constructable Element class

In the BIM process, much of the information required to perform a simulation study is defined and represented in a model. By using the resulting BIM model as input to generic parametric simulation models of construction activities, an integrated model of a construction project can be developed semi-automatically. With this approach, interdependencies between tasks, activities and resources are specified without a significant amount of manual work.

In the proposed framework for the integration of construction simulation and BIM, the adopted approach consists of assigning parametric simulation models of construction activities to individual BIM elements based on their type. The parametric models are generic in nature, since they can represent the construction process of a variety of construction products of the same type based on product and project-specific constraints. New instances of the simulation models of construction activities are created based on the product parameters provided by the BIM elements. These new simulation models are specific rather than generic, as they represent the construction activities required to build the BIM elements that provided the parameters to instantiate them. Figure 4.2 illustrates this process.

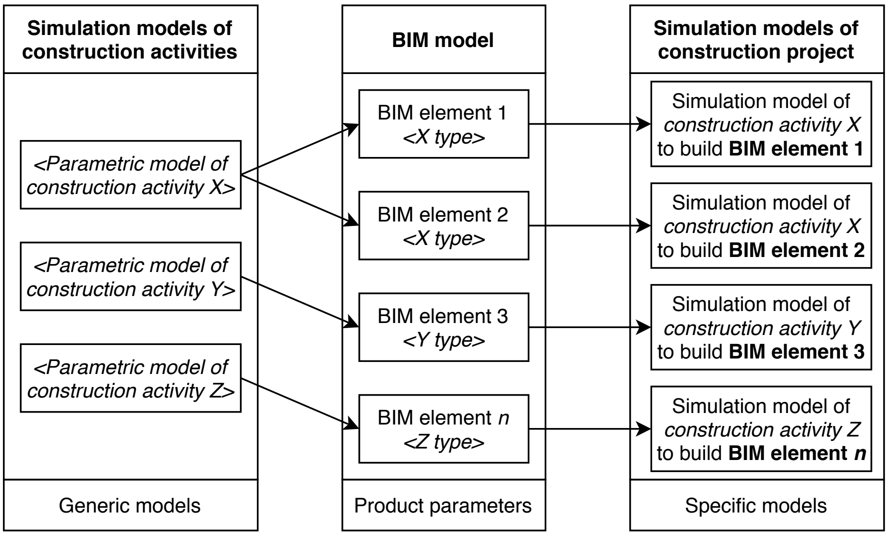


Figure 4.2: Assigning parametric simulation models of construction activities to BIM elements

Each of the parametric simulation models contains a control unit that determines its behaviour based on a set of rules that depends on the product parameters provided by the BIM element. Such control units are at the lowest level of the hierarchical control structure of the integrated model. The rules of the control units at higher levels of the hierarchy can be predefined by the user and can also be expressed in terms of product and project parameters. Furthermore, entity creation is also dependant on the product parameters. Figure 4.3 illustrates the integrated simulation approach adopted in the proposed framework.

The advantage of using a BIM model to provide the product parameters to the simulation models is that a lot of the required object definitions are already provided in the model. For example, in a BIM model, a column is clearly identified as an object of the column type, and its height and location within the project are determined. These parameters and definitions are often represented in the Industry Foundation Classes (IFC) format, which is a neutral, non-proprietary and open standard for sharing BIM data (König et al. 2012).

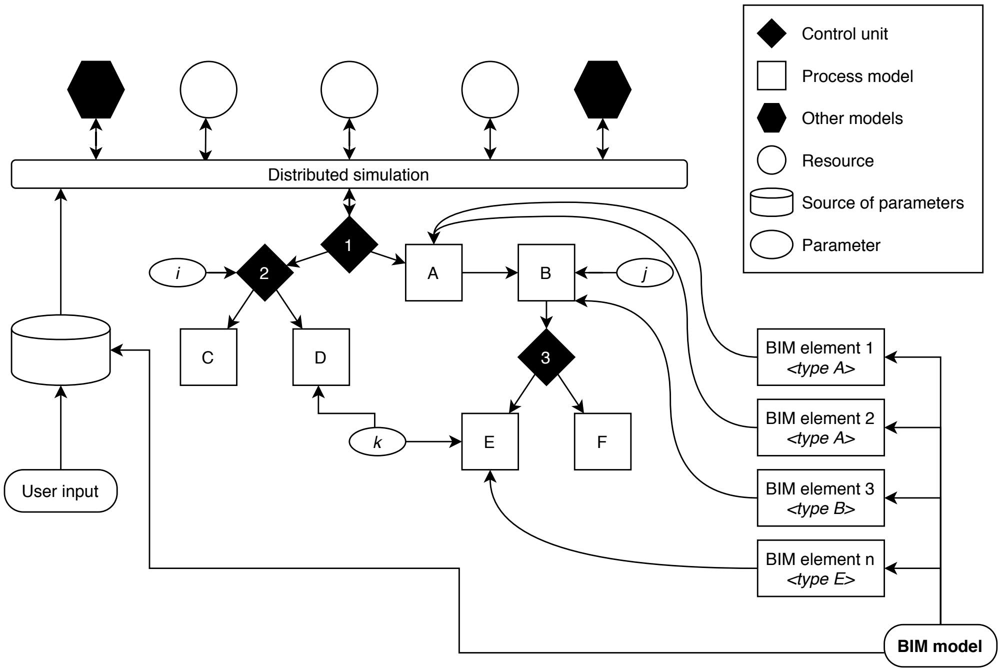


Figure 4.3: BIM-based parametric distributed simulation with hierarchical control structure

However, IFC is a rich and redundant data-modelling schema and clear definitions for implementation are required (S. Zhang et al. 2013). Moreover, temporary elements, which are frequently essential to consider in a simulation model such as formwork areas, are not generally defined in the BIM model (W. Lu and Olofsson 2014) or its corresponding IFC file. While the locations of BIM elements are defined in the model, the spatial relationships between them and site-related locations, such as the locations of material storage sites or work stations, are not defined in the BIM model either.

To overcome these limitations, an interface that translates BIM data into the requirements of the simulation models has been developed for the proposed framework. The interface, called *Constructable Element*, can extract BIM data from an IFC file or generate the parameters based on the geometry of a 3D model in a different format, for example, a filmbox (fbx) file. The purpose of this feature is twofold. First, it enables interoperability with non-IFC-based 3D modelling tools.

Second, it allows designers flexibility in their 3D modelling approach while still taking advantage of the framework.

The Constructable Element can be enriched to include required data that is not contained in the original BIM model, for example, formwork areas for concrete elements. This additional data can also be based on product or project parameters if required. The implementation of the interface is also based on the type of the BIM element. Figure 4.4 depicts the relationship between a BIM element, its corresponding Constructable Element, and its generic and specific simulation models.

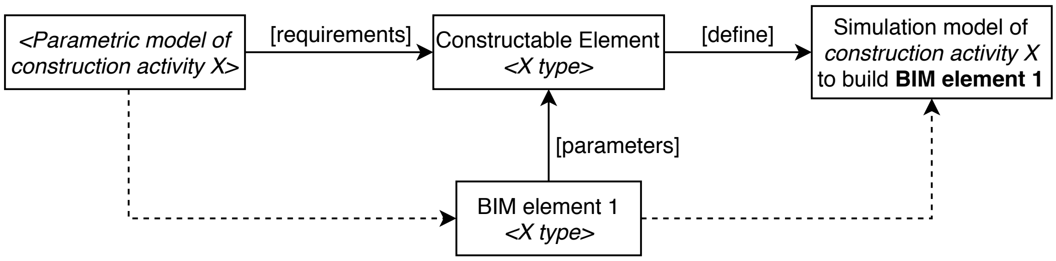


Figure 4.4: The Constructable Element interface

4.2.5 Animation of simulation results

In the proposed framework for the integration of construction simulation and BIM, a game engine is utilised to produce and visualise an animation of the simulation results. The approach used to generate the simulation-based animation is to adopt post-processing visualisation. Trace files that indicate the position and state of materials, workers and equipment during each step of the construction process of the simulated project are generated during each simulation run. A new trace file is published with the occurrence of each event in the simulation. Therefore, a collection of chronologically ordered trace files is available for each simulation run. Figure 4.5 depicts the concept of trace files in simulation-based animations.

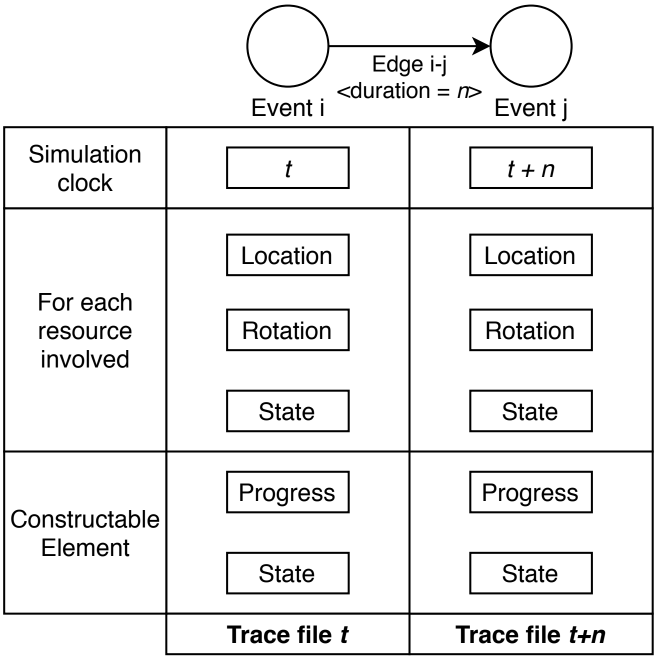


Figure 4.5: Trace file concept in simulation-based animations

The trace file concept is implemented following an object-oriented approach. Figure 4.6 depicts the trace file class diagram. When an event occurs during the simulation, the attributes of the game objects that represent the different resources involved in the event (materials, workers or equipment) are modified. At the same time, a new time-stamped trace file associated to each resource that changed is published. The new trace file saves the attributes of its corresponding resource at the corresponding simulation time. During the animation, the trace files are "played back" in chronological order. The game objects are located in the scene based on the attributes loaded from the trace file that corresponds to the animated time. A time-stamped trace file associated to a Constructable Element that changes is also published to track the progress of its construction as the model moves through time.

This approach allows users to create an animation with three different granularities or levels-of-detail, as illustrated in Figure 4.7. The first level-of-detail shows the Constructable Element at two distinct points during its construction, not con-

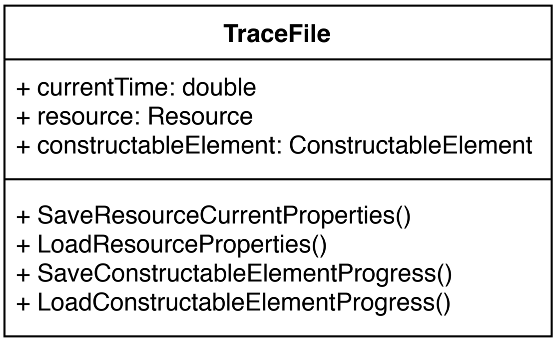


Figure 4.6: Trace file class diagram

structed and constructed. In order to create this animation, the trace file signals the game engine to switch the renderer off or on for the model of the element depending on its built status. The second level-of-detail requires different models of the Constructable Element at key stages of its construction. The trace file indicates the game engine to display the appropriate model at each stage during the animation. While this involves more modelling work, it may be relevant to present an animation at this level of detail to decision-makers during the planning process. The third level-of-detail shows the interaction between the different resources involved in the construction process of the element throughout the animation. In this case, the finalised model is displayed as the composition of its modelled materials. Although the trace files only contain the attributes of the resources at the discrete points in time in which events occurred, the continuous motion of the resources between two discrete moments could also be modelled using the game engine's functionalities.

The resulting animations can be visualised in a 3D environment that domain experts that are unfamiliar with simulation can easily interpret. Furthermore, by leveraging the game engine's compatibility with mixed reality technologies, the animations could also be visualised in an immersive virtual reality environment using a Head-Mounted Display.

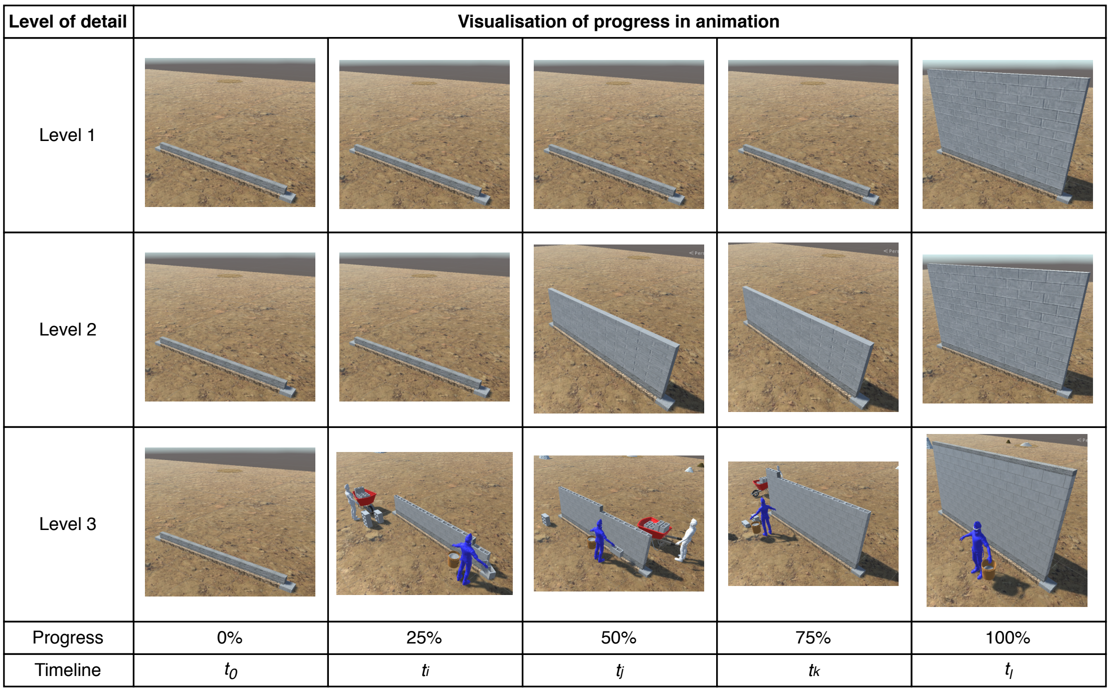


Figure 4.7: Visualisation of progress with different granularities of animations

Besides supporting the visualisation of simulation results, simulation-based animations have other contributions to the proposed framework for the integration of construction simulation and BIM, discussed in the following subsections.

4.2.6 Model verification and validation mechanism

In the context of simulation modelling, verification is defined as ensuring that the model matches the modeller's understanding of the system (Martinez 2009). On the other hand, validation confirms if the model possesses a satisfactory range of accuracy consistent with its intended application (Sargent 2013). In the simulation domain, it is widely accepted that though models cannot possibly be fully validated, it is useful to have some form of ensuring that the model is fit for its intended purpose, which is more important than model fidelity (Robinson et al. 2004). Simulation-based animations can serve both purposes (Kamat 2003; Sargent 2013). Modellers can view the animation to check the behaviour of the simulation model against its conceptual design to verify that it reflects their intentions. Domain experts that are not necessarily familiar with simulation can use the animation to validate the model.

4.2.7 Planning support

Simulation models are designed to help decision-makers make decisions or gain a better understanding of the modelled system (Robinson et al. 2004). Visualisation assists in investigating events that are hard to quantify in a definitive manner, but yet can affect the final outcome, for example, work zones overcrowding or safety hazards (Akhavian and Behzadan 2012). While 4D modelling is widely used for visualisation of construction, it only allows for the visualisation of construction schedules, remaining on the level of visualisation and not planning (Jeong et al. 2016). In this regard, the simulation approach utilised in this framework, coupled with the simulation-based animation at various levels-of-detail presented in this section can be exploited to support planning.

4.3 Environment

As shown in Figure 4.8, the environment module of the framework consists of three components: constructable elements, generic simulation models of construction activities and models of resources. These components enable the automatic generation of a construction simulation model based on a BIM model. 3D models of the different resources involved in the construction are also used to generate simulation-based animations.

Each component of the environment module of the framework can be seen as a container that stores loadable elements that the system uses to develop simula-

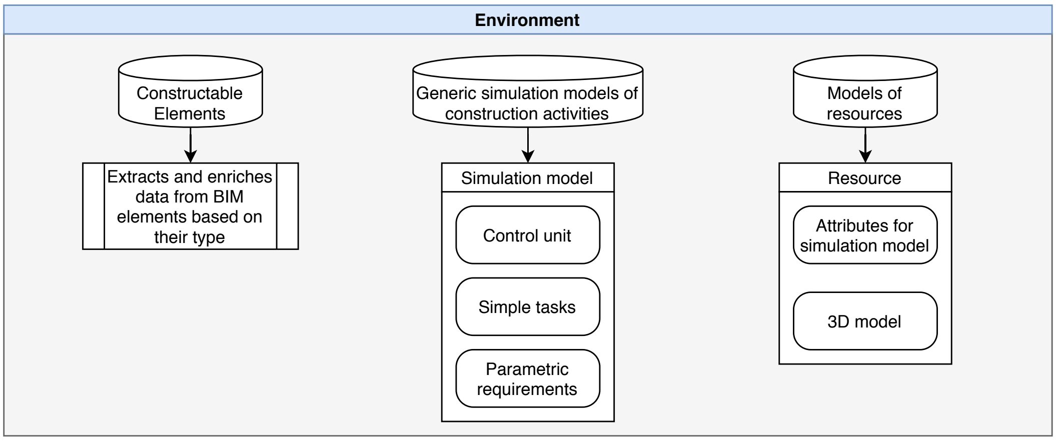


Figure 4.8: Environment components

tion models and animations. Once stored in their corresponding container, these elements can be reused as-is across multiple projects that involve similar construction products or resources. Each container can be expanded with new elements developed from scratch or adapted from existing elements to accommodate the user's requirements. As discussed earlier, the simulation approach of the proposed framework allows modellers to develop these elements at various levels of granularity and verify that they meet their requirements. The remaining of this section describes in more detail each component of the environment module of the proposed framework.

4.3.1 Constructable elements

The *Constructable Element* is an interface that extracts data from BIM elements and translates it into the requirements of the simulation models of construction activities. Furthermore, it can enrich the extracted data to include required parameters that are not contained in the original BIM model, but that can be computed from its product parameters, for example, formwork areas for a concrete element. The Constructable Element interface supports the Industry Foundation Classes (IFC) format. Additionally, it can also generate the required parameters based on the geometry of a 3D model in different formats, for example, an fbx file. This feature enables interoperability with non-IFC-based 3D modelling tools and flexibility in the 3D modelling approach of designers. Figure 4.9 depicts the information flow from a BIM element to its corresponding construction activity simulation model through the Constructable Element interface.

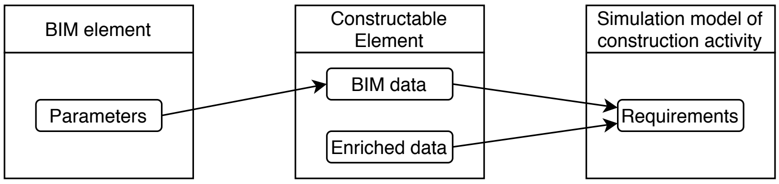


Figure 4.9: Information flow through the Constructable Element interface

The extracted and enriched data that the Constructable Element transfers to the generic simulation models of construction activities serve four purposes, discussed in the following paragraphs.

Determine the appropriate simulation model for the BIM element

The Constructable Element interface serves as a link between a BIM element and a simulation model of the construction activity required to build it. The Constructable Element assigns an appropriate simulation model from a library of models to the BIM element based on its type. When more than one simulation model exists for a given type of building element, additional parameters can be considered to support this feature. Alternatively, this process can also be done manually.

Create required entities

Entities in a simulation model are abstract representations of the objects that are processed in the system. In a typical construction activity, entities are closely related to materials. Based on the geometry of a construction product, the required amount of materials to build it can be computed. Therefore, the number of entities that the modelled activity needs to process to complete the simulation can be obtained from this type of parameter.

Change the control unit decision mechanism

Entity flow to tasks within each simulation model is controlled by a control unit. The decision mechanism of the control unit is based on a predefined set of rules. Some of these rules can be parameter based. For example, in a simulation model to fabricate and install the reinforcement of a column, the height of the column could be taken as a parameter that affects its tasks in different ways. The control unit of the federate could direct the entity flow to the task of overlapping rebar for transversal reinforcement if this parameter exceeded a certain length, for instance.

Establish interdependencies with other activities

One of the data that the Constructable Element interface generates from the geometry of an element's model is its topological relationship with other elements within the model. Based on these relationships and predefined rules at the top levels of the control structure hierarchy, the Constructable Element interface can automatically establish interdependencies between elements. For example, building elements located furthest from the material storage site could be built before or after those located closest. In another example, the construction of building elements that are supported by other building elements could begin as soon as the supporting elements are completed.

4.3.2 Generic simulation models of construction activities

This component of the environment module of the proposed framework consists of parametric simulation models that can represent the construction process of a variety of construction products of the same type based on product parameters and project-specific constraints. In this regard, these models are defined at the activity level as illustrated in Figure 4.10.

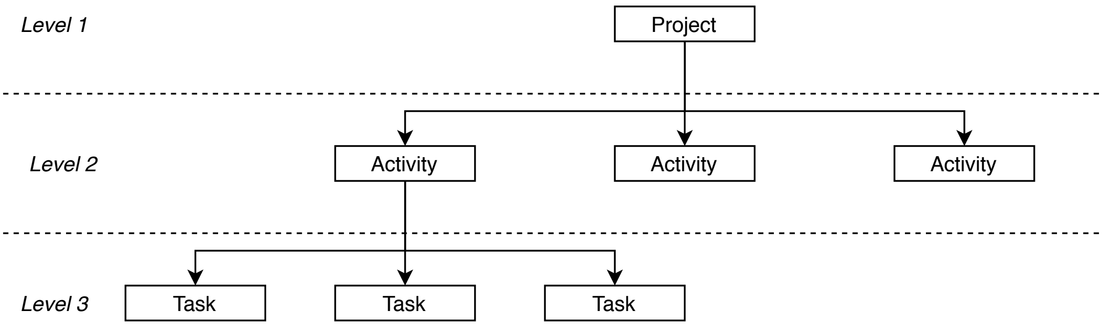


Figure 4.10: Typical hierarchy of a construction project

These activity models are based on the simulation modelling approach presented in the previous section. As such, they consist of a control unit linked to the simple tasks that compose the modelled activity. New instances of the simulation models are generated based on the parameters of each BIM element in the project. These instances are federated to form the construction project federation, following the distributed simulation paradigm. The subcomponents of the generic simulation models are described in the following paragraphs.

Simple tasks

As shown in Figure 4.10, tasks are the smallest unit into which a project is broken down. Within a construction activity, each task can take place zero, one or several times. They can also take place in sequential order, in parallel or in an overlapped fashion (Martinez 1996). When a task takes place, the state of one or more of the resources involved in it changes in some way over a time span defined as the duration of the task.

Each generic simulation model of a construction activity is composed of one or more tasks that specify which resources change and how when they take place. Additionally, each task has a duration that indicates by how much the simulation clock is advanced, as illustrated in Figure 4.11. Such duration can be either deterministic or stochastic. If the duration is deterministic, each time that the task takes place during simulation, the clock is advanced by the same predetermined amount of time.

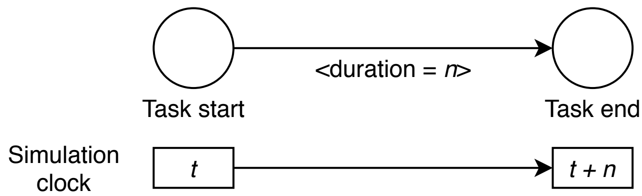


Figure 4.11: Duration of a task

On the other hand, if the duration of the task is stochastic, each time that the task takes place, the simulation clock is advanced by a different amount of time. This variation can be due to two different causes. First, the duration may be sampled at each occurrence of the task from a random variable with a predefined probability distribution. Second, the duration could depend on the state of the resources involved in the task at the time of its occurrence. For example, the duration of transporting materials between two areas of the construction site depends on the distance between the areas. The model can deal with such cases in two ways. First, the duration of the task can be expressed as a function that takes the condition that affects it as a parameter. While this is a straightforward workaround, if the dependent variable of the expression is random, which is when simulation models are more useful, it can lead to some degree of statistical error that reduces the fidelity of the model (Puri and Martinez 2013). Second, the modelled task can have more than one duration and take a sample from the most appropriate one based on the given conditions.

Control unit

Each activity simulation model has a control unit that controls entity flow within its tasks. The decision mechanism of the control unit of each model is based on predefined parameter-based rules to represent the logical relationships between the simple tasks that compose the modelled activity considering resource utilisation. These control units are automatically linked to control units in higher levels of the control structure hierarchy upon the instantiation of the activity simulation models to generate the simulation model of the construction project. Communication between control units at different levels of the hierarchy is both ways, meaning that the submodel of one activity can communicate with another submodel through the control unit at the level above them. This feature allows considering interdependencies between tasks of different submodels without re-formalising them.

Parametric requirements

As discussed previously, the generic simulation models take their required parameters from a BIM element through an instance of the constructable element interface. Such parameters determine the rules of the decision mechanism of the control unit of the submodel, the number of entities required to complete it and the duration of the tasks that depend on them.

4.3.3 Models of resources

Resources are things required to perform tasks (Martinez 1996). Resources in the framework are modelled following an object-oriented approach. Each resource object in the environment consists of the combination of two components, a 3D model that represents the resource in the simulation-based animation, and a set of attributes that represents it in the simulation model. In a construction activity, resources can be classified in three types, materials, workers and equipment. While all resources share some attributes, each type has some unique ones, as illustrated in Figure 4.12. The following paragraphs describe each of these resource types.

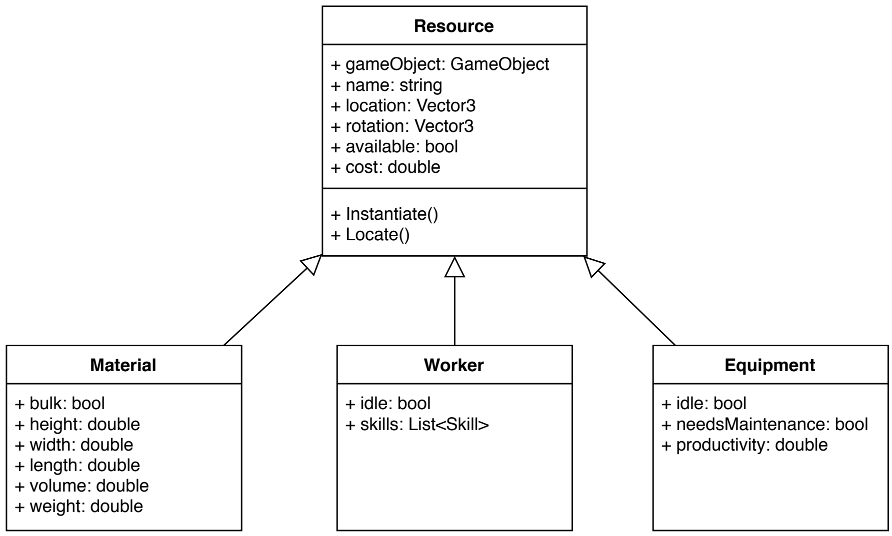


Figure 4.12: Resource class diagram

Materials

Elements of this type of resources are generally processed by workers, equipment or a combination of both. As previously mentioned, in a simulation model, this type of resource is closely related to the concept of entities. In this regard, the amount of materials that are required to complete the simulation model is computed from the parameters that the constructable elements extract from the BIM model. Materials are instantiated accordingly to represent this. It is worth mentioning that the simulation approach of the proposed framework allows modelling the procurement process of materials as an independent model that constrains their availability in the integrated model.

In construction, materials are further classified as discrete or non-bulk materials, such as bricks or reinforcement bars, and bulk or non-discrete materials, such as water or cement. Materials in the environment module of the proposed framework can represent both types. In some instances, bulk materials can be handled as discrete. For example, when mixing concrete on site, cement can be measured in terms of bags. In that case, one bag of cement is a discrete material.

Workers

Workers in the framework have two unique attributes. First, their state during simulation, which can be either idle or busy. Workers are busy while they are performing a task, and idle while they are not. Second, the list of skills that they have. Workers can have one or more skills, each of which makes them eligible to perform different tasks during the simulation. When a task is ready to be executed, if the required number of workers that have the required skills to perform it is idle, they are assigned to the task and their states change to busy.

Equipment

Generally, each piece of equipment is specifically designed by the manufacturer to perform specific operations (Schaufelberger and Migliaccio 2019). An attribute similar to the workers' list of skills is not appropriate for this type of resource. Instead, each instance of this class should represent a specific type of equipment that the simulation model should explicitly demand. On the other hand, the state of the equipment is similar to that of workers (busy or idle) with the addition of a third state (requires maintenance), which can change based on its utilisation time or any other rules imposed by the modeller. The simulation approach of the proposed framework allows modelling equipment breakage and maintenance as independent models that constrain equipment availability in the integrated model.

Combination of different types of resources

Combinations of different types of resources are too common in construction. For example, a worker transporting a wheelbarrow filled with fine aggregate is composed of the combination of a worker, a piece of equipment (the wheelbarrow) and a quantity of bulk material (the aggregate). The object-oriented approach to modelling resources enables these combinations to occur during simulation without modelling workarounds.

4.4 User input

To build a simulation model that represents the uniqueness of a construction project, it is essential to input data that accurately describes such uniqueness. BIM models, which have been widely adopted by the construction industry, provide information that can be leveraged to semi-automatically develop a simulation model. In addition to this, the proposed framework enables users to explore scenarios under different resource availability and other project-specific constraints. With that in mind, this module of the proposed framework is composed of the three components depicted in Figure 4.13. The remaining of this section describes in more detail each of these components.

4.4.1 BIM model

A BIM model stores a significant amount of information of a building. For the purpose of construction simulation, only certain properties of the BIM elements

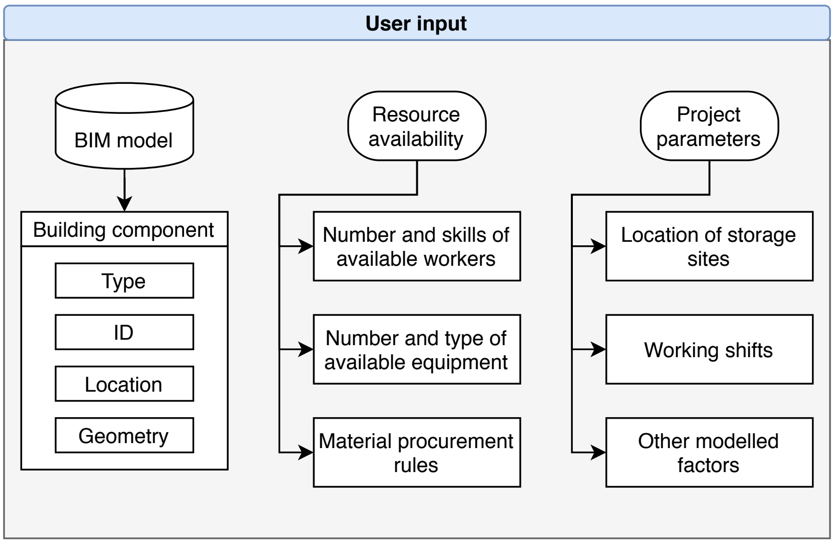


Figure 4.13: User input components

are relevant. As described previously, an instance of the constructable element interface extracts from each BIM element the relevant information required by the simulation model based on its type. Such information can be enriched to provide the simulation with data that is not necessarily included in the original BIM model. Therefore, the level of information that each BIM element should contain is based on the requirements of the relevant simulation model. However, there are some product parameters that should be considered in most cases. These parameters are identified and discussed in the following paragraphs.

Building component type

The simulation model of the construction activity required to build the BIM element is assigned based on the building component type. Therefore, the building component type must be clearly defined for each BIM element that will take part of the construction project federation. In this regard, the building component type makes reference to the data hierarchy of Revit, a popular BIM software, as illustrated in Figure 4.14.

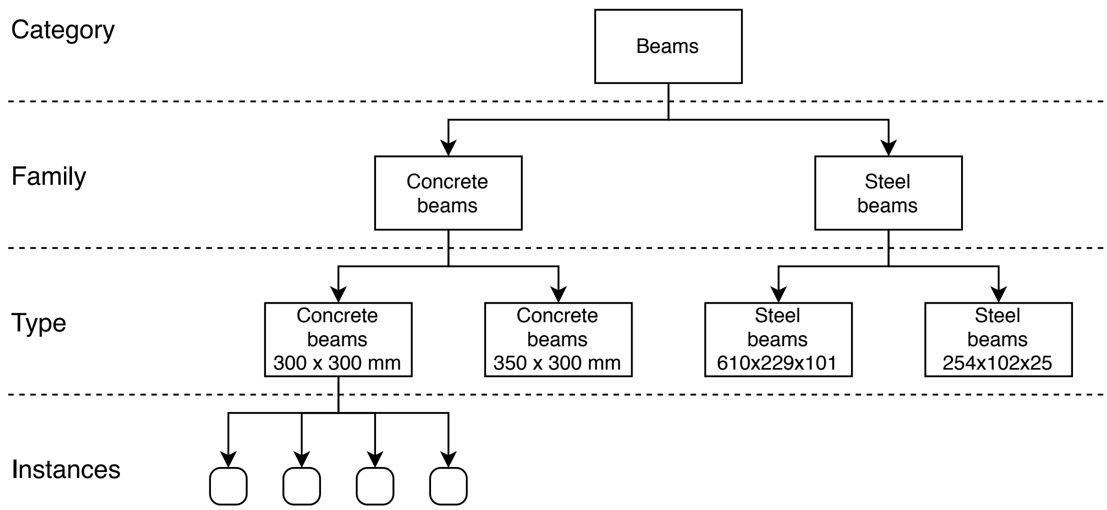


Figure 4.14: Revit data hierarchy

The building component type of a BIM element not only indicates the element's type of component, but also some common characteristics usually related to materials or standard geometry. These characteristics govern all the unique instances of the same type in the model.

ID

To differentiate between different instances of the same element type, BIM elements usually have a unique identifier. This identifier is the name of the unique building component that the element represents.

Location

This parameter refers to the coordinates that place the BIM element in a specific location in reference to a known base point. Coupled with the geometry of the element, this datum is useful to identify the topological relationships of the element to other building components. Moreover, spatial relationships with other relevant known locations within the model, such as storage sites or related workstations, can also be established.

Geometry

Besides facilitating the visual representation of the building components in the 3D model, their geometry also allows the calculation of the material quantities required to build them. Thus, this parameter is typically required as input for the corresponding simulation model. Naturally, the required level of detail of the modelled geometry depends on the requirements and granularity of the simulation model.

4.4.2 Resource availability

As described in Section 4.3, resources in the proposed framework are loadable components separated from the simulation model. In fact, they are modelled as independent federates that also subscribe to the federation. To represent resource availability and constraints in the simulation, users can define the parameters discussed in the following paragraphs.

Workers

Users set up the number of available workers in the model. New instances of the worker type with unique identifiers will be created in the environment based on this number. Users can define how many and what skills each instance of the workers has. An area within the construction site can also be defined to locate idle workers for visualisation purposes.

Equipment

Users set up the number and type of each piece of equipment available in the model. An initial location can be defined for each piece of equipment as well as an area to locate each idle instance of equipment.

Materials

As mentioned in Section 4.3, the quantities of materials required to build the input model is computed from the parameters extracted from the BIM model and enriched by the Constructable Element interface. For this reason, quantities of materials are generally not subject to user input. However, users can define the locations and capacities of the storage sites of each type of raw material. Restock of such storage sites is automatic by default, but a time-stamped informative report is provided as output. Additionally, the simulation approach of the proposed framework allows users to include independent federates that represent the procurement process of materials.

4.4.3 Project parameters

Users can input parameters that affect the decision mechanism of control units at the top levels of the control structure hierarchy, such as working shifts, limits on the number of workers in certain areas of the site, etc. Additionally, users can override the automatically generated interdependencies between federates by explicitly determining a construction sequence or scheduling specific activities.

In addition, users can input independent federates to represent other project-specific constraints. The simulation approach adopted to design the proposed framework allows users to include independent federates of factors that affect the semi-automatically developed construction simulation model. Examples of these factors are labour absenteeism, equipment breakage and maintenance, materials procurement processes, safety inspections or location-based weather.

4.5 Pre-processing module

The purpose of this module of the proposed framework is to semi-automatically develop the construction simulation model of a construction project based on its BIM model and predefined generic simulation models of construction activities. To do that, it instantiates a specific simulation model of the required construction activity to build each element of a BIM model. These specific models are instantiated by assigning values to the parameters of a corresponding generic simulation model. Such values are extracted from the BIM elements by an instance of the Constructable Element interface. The remaining of this section describes in more detail this process and the requirements of the module.

4.5.1 Inputs and outputs

Figure 4.15 illustrates the inputs and outputs of the preprocessing module of the proposed framework. As the figure shows, in this module only the BIM model is input by the user since the Constructable Elements and generic simulation models are preloaded in the environment module of the framework, as detailed in Section 4.3.

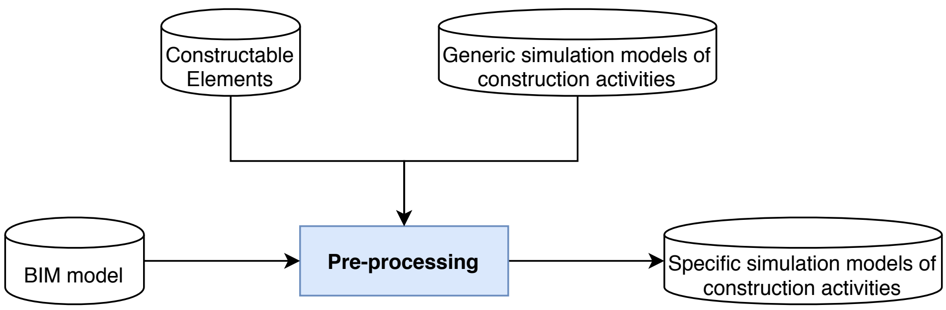


Figure 4.15: Inputs and outputs of the pre-processing module

The outputs of this module are the specific simulation models of the construction activities required to build the BIM elements from the model. These simulation models are added to the control structure of the distributed simulation model that represents the construction process of the entire project. By doing this, the construction activity federates form the construction project federation, which can be simulated in the simulation module.

4.5.2 Instantiating simulation models of construction activities

Figure 4.16 depicts the flow diagram of the pre-processing module of the proposed framework. As the figure shows, the process to instantiate the specific simulation models of the construction activities is automatic. However, users can override this process manually before the simulation begins to accommodate their needs.

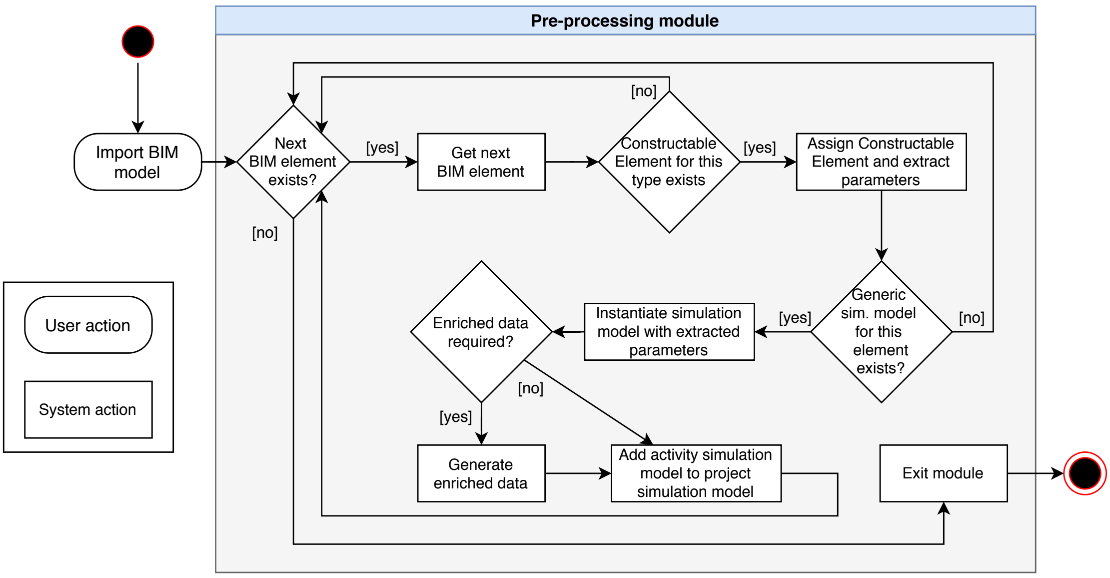


Figure 4.16: Flow diagram of the preprocessing module

4.6 Simulation module

As shown in Figure 4.17, the simulation module of the proposed framework is composed of six components: a distributed simulation, a construction project simulation model, resources simulation models, modifying parameters, other simulation models and trace files. The first five components make up the conditions under which the simulation is carried out, while the latter one produces the results for visualisation and analysis. This module and its components are designed based on the requirements outlined in Section 4.2.

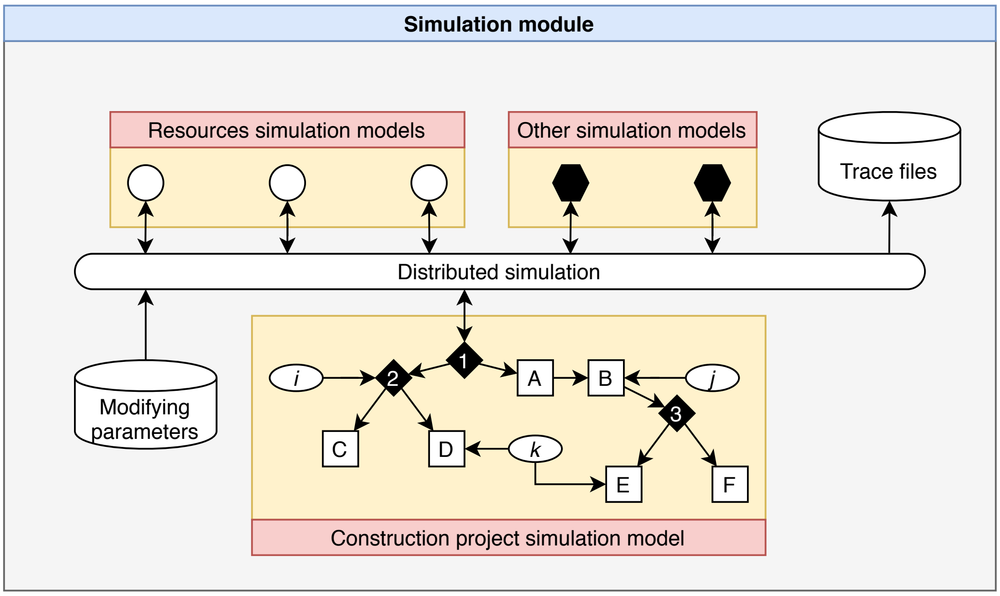


Figure 4.17: Simulation module components

The remaining of this section describes in more detail each component of the simulation module of the proposed framework and how they interact with each other.

4.6.1 Distributed simulation

The distributed simulation component is the central component of the simulation module of the proposed framework. As such, it interacts with all the other components of the module in different ways. Moreover, it contains the required infrastructure to run the simulation and produce its results. The simulation clock is embedded within this component and it is advanced by it based on requests from other elements of the module.

Interaction with other components

The distributed simulation component interacts with all the other elements in the module in three different manners: read-only, write-only and read and write, as illustrated in Table 4.1.

Table 4.1: Types of interactions between the distributed simulation component and other elements in the simulation module

Interaction type	Characteristics	Receiver components
Read-only	Extract information from receiver component without updating its data	Modifying parameters
Write-only	Update data of receiver component without extracting data from it	Trace files
Read and write	Extract and update data of receiver component	Construction simulation model Resources simulation models Other simulation models

In a read-only interaction, the distributed simulation component is able to extract information from another element without being able to update that element. This type of interaction occurs with the modifying parameters input by the user. The distributed simulation component reads these parameters and updates other elements in the module based on them. On the other hand, in a write-only interaction, the distributed simulation component updates data of another component without extracting any information from it. This type of communication happens with the trace files, which are generated (or *published*) by the distributed simulation component based on information from other components of the module.

The read and write type of interaction refers to two-way communication. In this type of interactions, the distributed simulation component is able to extract information from another element as well as to update its data. This type of interaction occurs between the distributed simulation component and the construction simulation models of construction activities, the resources simulation models and the other simulation models in the module. In other words, the central component of this module communicates in this way with all the simulation federates of the federation.

The remaining components of the module can communicate with each other through the distributed simulation component. None of them is able to directly update the information of another element; however, they can request that the distributed simulation component do so. For example, a construction activity federate model may request the relocation of a resource so that one of its tasks can be carried out. The distributed simulation component reads the resource information and, if the resource is available, it updates its location and status to represent that the resource has been assigned to the required task. Equally, none of the other components of the module can directly change the simulation clock.

Time advancement and model update mechanism

Whenever a task is due to take place, its parent federate sends a request to the distributed simulation component of the module to schedule the task. The distributed simulation component checks if all the conditions for executing the task are met. If that is the case, the resources required to perform the task are assigned to it and the task is added to a list of scheduled tasks alongside its expected duration.

Figure 4.18 depicts how the distributed simulation component of the simulation module advances the simulation clock and updates the properties of the federates based on the scheduled task with the earliest finish time. The simulation clock is advanced to the finish time of the scheduled task that has been executed.

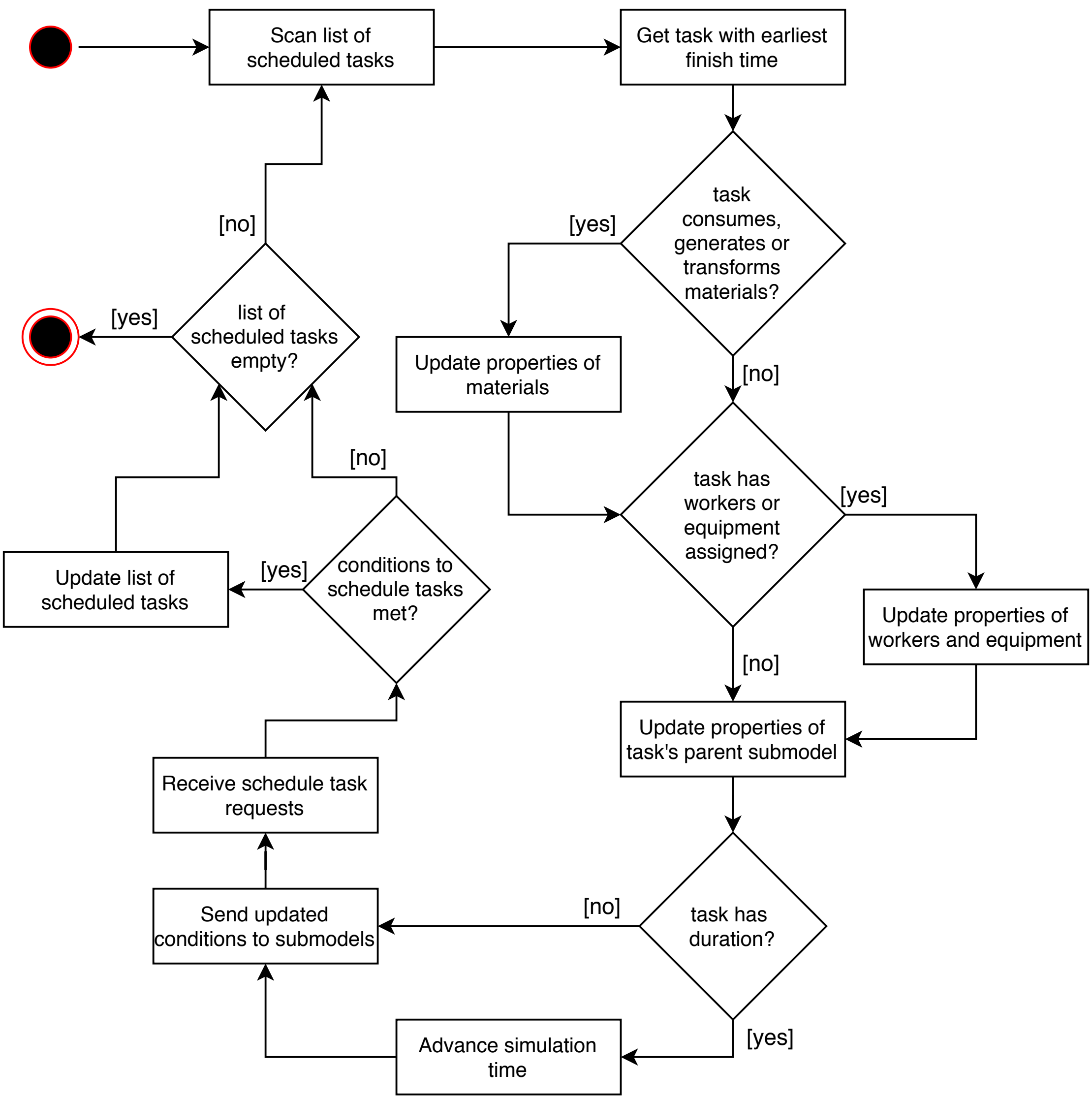


Figure 4.18: Time advancement flow diagram

4.6.2 Construction project simulation model

This component of the simulation module is composed of the specific simulation models of construction activities to build the elements of the BIM model imported by the user. These models are automatically generated in the pre-processing module of the proposed framework, as detailed in Section 4.5. As previously described, each of these models contains a control unit and the set of simple tasks that make up the modelled construction activity. The control units of the models are linked to control units in higher levels of the control structure hierarchy.

The control unit at the highest level of the structure communicates with the distributed simulation component before and during the simulation. Before the simulation begins, the control unit gets information regarding the modifying parameters input by the user. Consequently, the parameter-based rules of the control units located downstream in the control structure hierarchy are adjusted accordingly.

During the simulation, the control unit at the highest level of the control structure hierarchy gets updates from the distributed simulation component regarding the current state of the project and resource availability. Subsequently, it sends signals downstream to the control units in the lower levels of the hierarchy based on its predefined behavioural rules until all the conditions to execute one of the tasks of one of the models are met. In this regard, control units act as valves that allow or prevent information flow to relevant submodels.

Once a task is ready to be carried out, a request is sent upstream to the distributed simulation component through the control structure hierarchy to acquire the required resources and schedule the task. Such request includes the value of the duration of the task, which, as detailed in Section 4.3.2, can be either deterministic, sampled from a probability distribution or computed from a parameter-based expression. As previously discussed, the distributed simulation component updates the simulation clock, the resources information and the status of the project accordingly. This process is repeated until the completion of the project.

4.6.3 Resources simulation models

As described in Section 4.3.3, the attributes of the different simulated resources are predefined in the environment module of the proposed framework. During simulation time, several instances of the different resources involved in the construction project are available to participate in the tasks of the construction simulation model. The properties of each resource instance are available for reading by the distributed simulation component. These properties are also available for updating by the distributed simulation component based on requests generated by the tasks that utilise the resources. The following paragraphs describe the different ways in which resource models change during simulation based on their type.

Workers and equipment

Based on user input, resources of the worker and equipment types are instantiated and available during the simulation. Generally, these types of resources are needed to process materials in a construction task. Whenever all the conditions that are not related to these types of resources are met for a task of the simulation model to take place, a request to assign the required resources is sent to the distributed simulation component of the module. If the resources are available, they are assigned to the task, which is scheduled to be performed, as illustrated in Figure 4.19. This means that the statuses of the resources are changed from idle to busy and that the resources are associated with the scheduled task until it is finished.

One of the conditions that are not related to workers or equipment resources that has to be met to schedule a task is material availability on site, discussed below.

Materials

Before the simulation begins, based on the quantities of materials required to build the BIM elements in the model computed by the corresponding constructable elements, resources of the material type are instantiated in the simulation model.

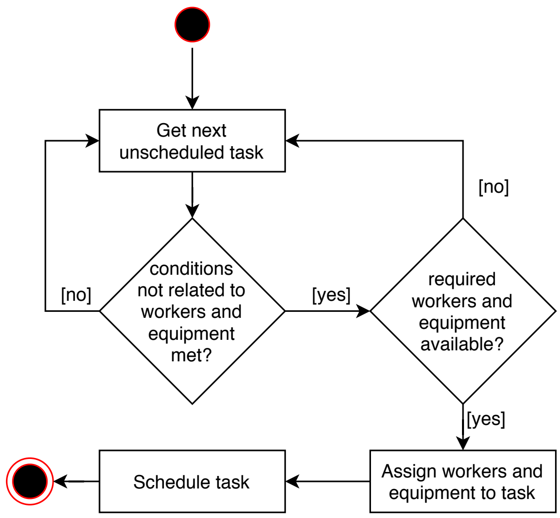


Figure 4.19: Workers and equipment assignment

These initial instances represent the raw materials required during construction. Such materials are processed by workers and equipment during the simulation as the tasks that utilised them are performed.

Material processing can result in different types of changes in the properties of the processed materials. For example, if a material is transported from one location to another, its location property is updated. If materials are transformed into new different materials, the amount or number of the source materials decreases while the amount or number of the new materials increases. The last case may involve instantiating new materials during simulation time. As the distributed simulation component of the module updates these properties, submodels that require the materials request them so that their tasks can be scheduled.

4.6.4 Modifying parameters

As discussed in Section 4.4, these parameters are either direct user input or based on features extracted from the user input BIM model. Some of these parameters are passed to the relevant construction activity simulation models in the pre-processing stage, as discussed in Section 4.5.

In this module, the distributed simulation component extracts the relevant modifying parameters and uses them to update the parameter-based rules of the control units in the higher levels of the control structure hierarchy of the construction simulation model. This process occurs only once, before starting the simulation. However, the process is not part of the preprocessing stage because modifying these parameters does not lead to instantiation of new simulation models of construction activities. Instead, it modifies the behavioural rules of their control units.

4.6.5 Other simulation models

Based on the distributed simulation approach adopted in this research, other simulation models can run concurrently with the construction simulation model. These federates are subscribed to the distributed simulation component and communicate with it in a similar fashion than the construction simulation model federate. They can also send requests to acquire available resources, schedule tasks, advance the simulation clock or interrupt scheduled tasks.

Figure 4.20 illustrates a labour absenteeism simulation federate interacting with the distributed simulation component. This model instantiates a new labour absence and requests a worker to "execute" it. The distributed simulation component selects a random worker regardless of its status. If the worker is idle, the absence is immediately scheduled with the now busy worker assigned to it. If the worker is busy, the task to which the worker is assigned is interrupted and the other resources involved in the task are released. The status of the worker is set back to idle once the duration of the absence is met.

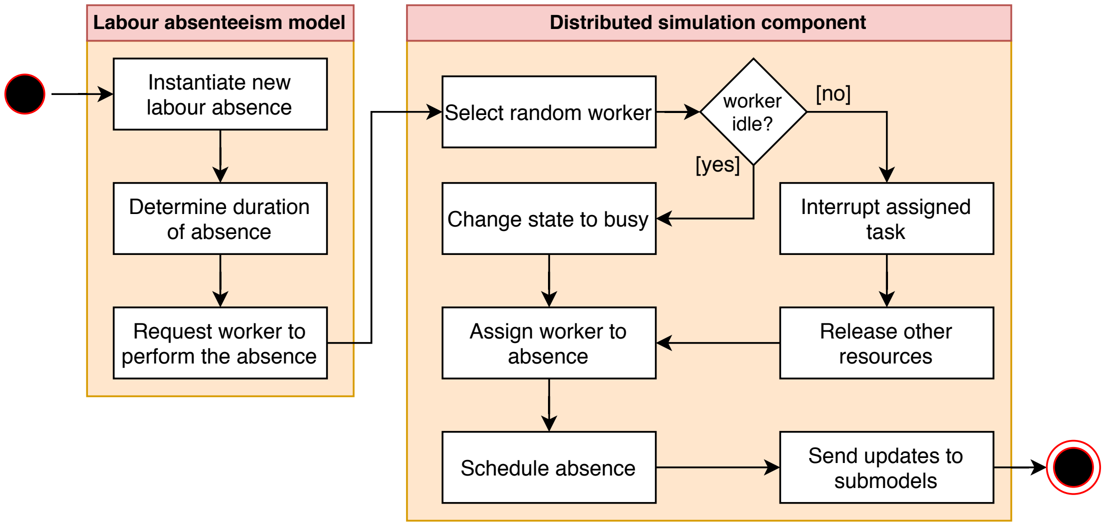


Figure 4.20: Labour absenteeism simulation model interacting with the distributed simulation component

4.6.6 Trace files

This component of the simulation module stores the information required to visualise and analyse the simulation results. During each simulation run, the distributed simulation component publishes a new time-stamped trace file every time that there is an update on the system. Essentially, the chronologically ordered set of trace files of a given simulation run is a detailed log of all the events that occurred during the simulation. Such sets of trace files include all the required information to produce reports on resource utilisation and task durations.

Equally, the trace files store the values of the properties of the resources involved in the construction throughout the simulation time. This information can be utilised to produce a simulation-based animation for different visualisation purposes, including model verification and validation, planning and decision-making support and communicating results to stakeholders unfamiliar with simulation. The following section further discusses the utilisation of trace files to visualise and analyse the simulation results.

4.7 Visualisation module

As discussed in Section 4.6.6, the simulation module of the proposed framework produces a set of time-stamped trace files for each simulation run of the integrated simulation model. Such trace files contain relevant information that can be used to support planning and decision-making in the earlier stages of the construction project management process. The visualisation module of the proposed framework displays the simulation outputs in different ways, classified into two main categories: reports and animations, as depicted in Figure 4.21.

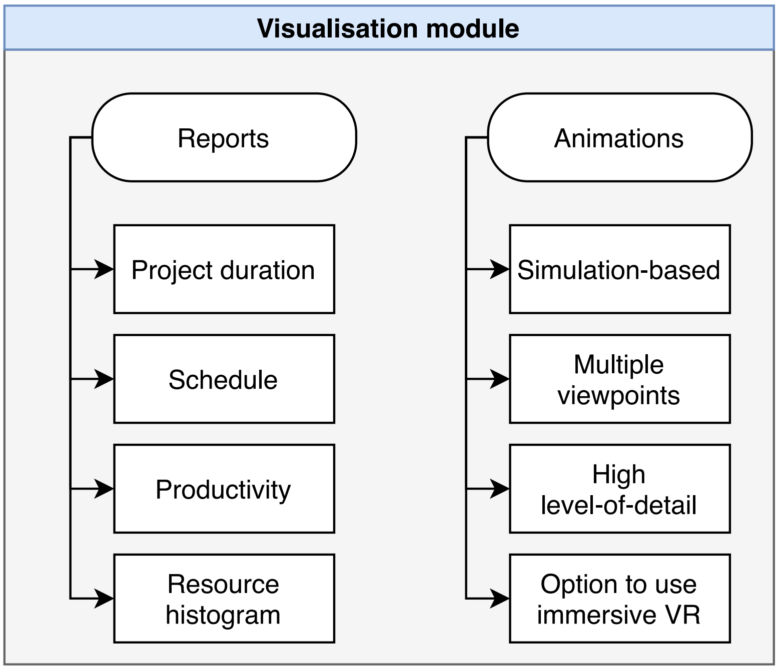


Figure 4.21: Simulation results categories

The remainder of this section describes the categories into which the simulation outputs are classified in more detail.

4.7.1 Reports

While a set of trace files generated in the simulation module contains a large amount of useful information for construction managers, these raw data need to be organised in order to support informed decision-making properly. This is especially true when the durations of the tasks are not deterministic since more than one simulation run is necessary to obtain reliable results to make predictions on the performance of the project. In this case, more than one set of trace files becomes available to analyse, resulting in an immense amount of data.

In the visualisation module of the proposed framework, the data contained in the trace files can be displayed in four different types of reports, discussed in the following paragraphs.

Project duration

When the durations of the tasks of the modelled construction activities is stochastic, each simulation run results in a different project duration. The last trace file of the set of trace files that corresponds to a simulation run indicates the project duration in that run. The project duration report shows a frequency histogram of the resulting project duration in each simulation run. According to the central limit theorem, if the number of simulation runs is greater than thirty, the data set of durations will be approximately normally distributed. The parameters of the normal distribution (mean and standard deviation) that adjusts to the project duration data set can be used to determine the probability of completing the construction project on a given date. This information is particularly useful during construction planning, as it may reveal the need to make changes to the construction strategy to meet the project deadlines.

Schedule

A schedule can be automatically generated for each simulation run using its corresponding data set. During the planning stage of the construction project, two or more of these schedules can be compared to support decision-making. Users can select which schedules they want to visualise based on the probability that their total duration is met, as described above.

Productivity

Productivity is an important metric in construction. Using the trace files, productivity can be computed for each simulation run at three different levels using Equation 4.1 (Project Management Institute 2013).

$$P_{ij}^{k} = \frac{Q_{ij}}{D_{ij} * N_{ij}} \tag{4.1}$$

where P_{ij}^{k} is the productivity, Q_{ij} is the total amount of work executed, D_{ij} is the duration of the execution of the work and N_{ij} is the number of resources involved in the work.

At the project level, one productivity value is calculated for each simulation run. At this level, Q_{ij} is considered as one unit of the simulated project, D_{ij} is the total duration of the simulation run and N_{ij} is considered as one unit of workforce and equipment.

At the activity and task levels, several productivity values are computed for each simulation run. At the activity level, a productivity value is calculated for each submodel of each construction activity for each simulation run. Q_{ij} is computed from the corresponding BIM element parameters extracted by the constructable element interface (e.g. the side area of a concrete block wall), D_{ij} is the total duration of the construction activity and N_{ij} is considered as one unit of workforce and equipment. Similarly, a productivity value can be calculated whenever a task takes place. At this level, Q_{ij} is considered as the amount of production units of the task (e.g. volume of mortar mixed, concrete blocks carried from the storage site to the wall construction site or concrete blocks laid), D_{ij} is the total duration of the task and N_{ij} is considered as one unit of workforce and equipment.

At each level, this information is useful for different purposes. At the project level, it can be used as a benchmark for similar projects. At the lower levels, productivity can be useful to identify bottlenecks in the model and help decision-makers to implement measurements to prevent productivity loss in the real system.

Resource histogram

Based on resource utilisation during each simulation run, type-based resource histograms can be plotted to support resource scheduling and levelling during construction planning. This information is useful to identify long periods in which workers or pieces of equipment are idle, as well as the needed frequency of restocking of the different materials.

4.7.2 Animations

Simulation-based animations can be used for model verification and validation, as a mean of communication with stakeholders unfamiliar with simulation and as a tool to support planning and decision-making. Game engines can be leveraged to produce simulation-based animations.

The visualisation module of the proposed framework produces a simulation-based animation for each simulation run. Such animations take as input three elements, as depicted in Figure 4.22. First, the 3D model subcomponent of the resources instantiated during simulation. These 3D models are initially loaded into the environment module, as discussed in Section 4.3.3. Second, the geometry of the BIM elements of the BIM model input by the user, as discussed in Section 4.4.1. Third, the set of trace files corresponding to the simulation run of interest. This set of trace files is generated in the simulation module, as discussed in Section 4.6.6.

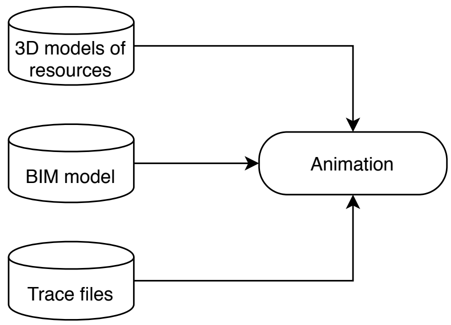


Figure 4.22: Inputs to produce a simulation-based animation

The adopted approach of the proposed framework to produce simulation-based animations allows producing animations at three levels-of-detail. Figures 4.23, 4.24 and 4.25 depict the process to prepare an animation at level-of-detail one, two and three, respectively.

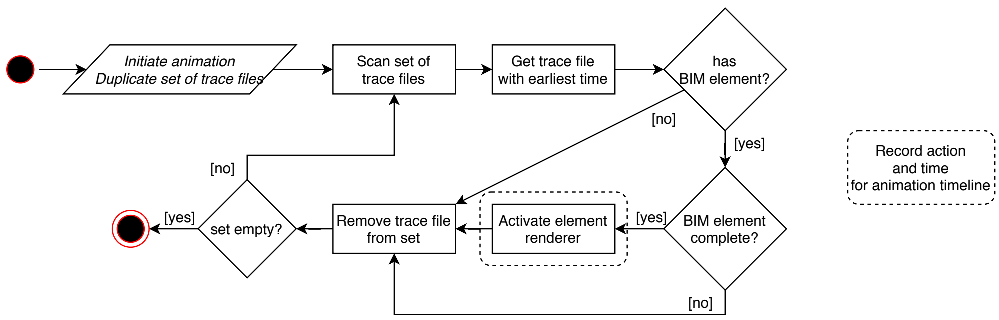


Figure 4.23: Preparing a simulation-based animation at level-of-detail 1

Once the set of trace file has been processed as shown in the figures, the simulation-based animations can be replayed and visualised from multiple viewpoints. Fur-

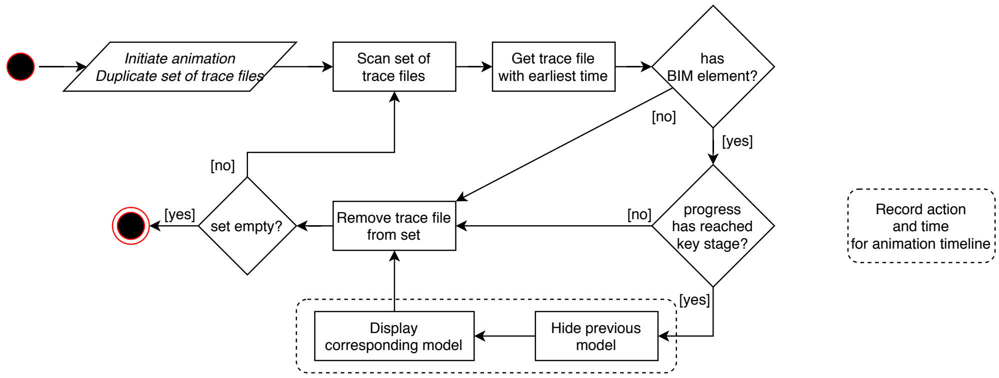


Figure 4.24: Preparing a simulation-based animation at level-of-detail 2

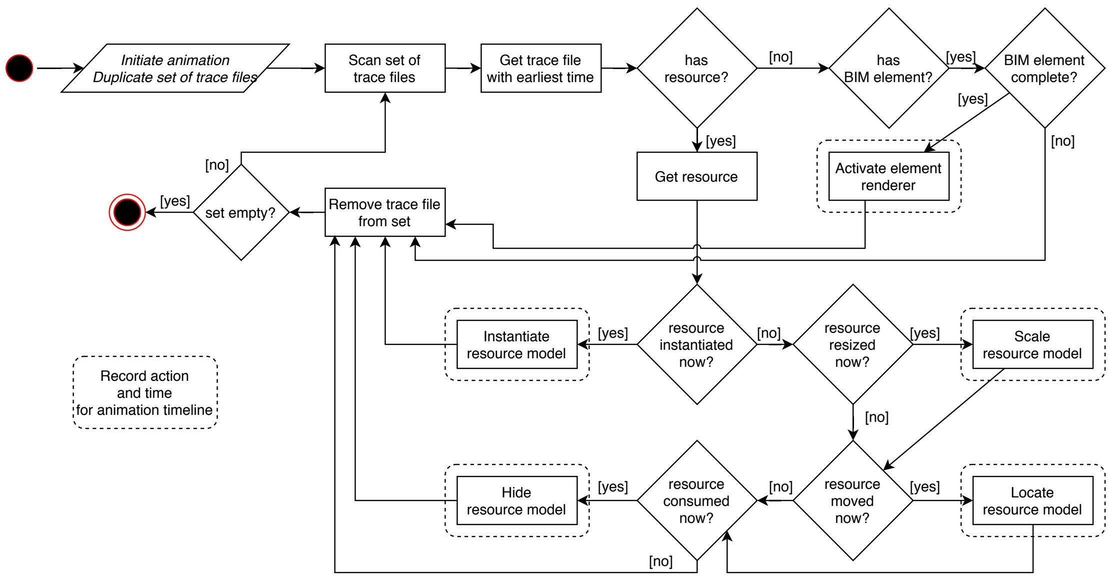


Figure 4.25: Preparing a simulation-based animation at level-of-detail 3

thermore, leveraging the functionalities of game engines, the animations can also be visualised in an immersive virtual reality environment, which has shown a great potential to improve workflow efficiency in the construction context through enhanced common understanding (Du et al. 2018).

At the third level-of-detail, the simulation-based animations show resource interaction in detail as the construction process progresses. A current limitation of this approach is that it only shows the location of resources at the discrete points in time in which events occurred and triggered the instantiation of a trace file. Thus, the continuous motion of resources is not available to detect, for example, intersections of moving routes in the construction-site.

However, the source and target points, as well as the start and finish times of the motion are available in the trace files. Therefore, there is potential to overcome this limitation by using these data to model the missing continuous motion and support clash detections. Notably, this can be achieved without redefining submodels discretisation to an impractical higher granularity, such as modelling every step that a worker takes to move materials.

4.8 Summary

This chapter provided an overview of the proposed framework to integrate construction simulation and building information modelling, and described in detail its five main modules. It described the system requirements to develop a system based on the proposed conceptual framework. Moreover, it discussed how the components of each module respond to the adopted simulation approach. The chapter also described the required user inputs to semi-automatically develop a construction simulation model based on an existing BIM model. It additionally addressed the uses of the outputs of the simulation in construction planning, resource scheduling, decision-making, model verification and validation and as a mean of communication with stakeholders unfamiliar with simulation. The next chapter covers the implementation of the proposed framework within a game engine.

Chapter 5

Implementation of the proposed framework within a game engine

The previous chapter introduced a framework for the integration of construction simulation and building information modelling (BIM). In addition to exploiting simulation and BIM technologies, the proposed framework leverages gaming technologies, specifically a game engine, to produce and visualise simulation-based animations.

This chapter describes the implementation of the proposed framework for the integration of construction simulation and BIM within a game engine. An overview of the game engine concepts utilised in the presented implementation is introduced first. Then, the chapter details the implementation of each module of the proposed framework within the game engine in the same order in which they were introduced in Chapter 4.

This chapter illustrates many of the components of the proposed frameworks using Unified Modelling Language (UML) class diagrams. Appendix A shows a table with the notation used in such diagrams.

5.1 Game engines

In recent years, game engines have drawn the attention of researchers and practitioners in the architecture, engineering and construction industry. Game engines provide the complex technological foundation upon which digital games are built (Anderson et al. 2013). They are commonly implemented as an integrated collection of modules with different functionalities, for example, graphics generation, physical behaviour of objects, collision detection, user interaction, sound management, etc. (Petridis et al. 2010). Game engines provide ways to achieve immersion in visualisation due to their compatibility with mixed reality technologies, ubiquity as they offer multi-platform deployment, high-quality graphics and network capabilities that allow for multiple simultaneous users, among other characteristics (Osorio-Sandoval, Tizani and Koch 2018).

While there are several commercial off-the-shelf game engines available in the market, researchers that use them tend to have a preference for Unity, as revealed by a literature review on serious games with focus on technology application. Amongst the reasons for choosing Unity over other game engines, the most relevant appear to be functionality, multi-platform deployment, availability at no cost and capability of interacting with major 3D tools and file formats (Osorio-Sandoval et al. 2017), including 3D models authored in Autodesk Revit, a popular building information modelling software (Bille et al. 2014; Dib and Adamo-Villani 2013; W. Wu and Kaushik 2015). For these reasons, the Unity game engine was selected as the game engine to implement the proposed framework for the integration of construction simulation and BIM. However, the framework can be implemented in other platforms due to its generic conceptualisation.

In the remainder of this section, some key concepts related to the game engine utilised in the implementation of the proposed framework are described.

5.1.1 Scene, game objects and game components

Within a game engine, game objects are the fundamental objects that represent characters, props and scenery. They can be seen as containers for the game components, which implement the real functionality (Unity Technologies 2019). The scene is what manages all of the game objects. It receives updates and passes them onto the components attached to game objects in the scene (Lambert 2014) to dictate their behaviour.

Most game engines have a set of several predefined types of game components with different specific functionalities, and a single game object can have multiple of them attached. For example, a simple solid object in a game scene requires at least four game components: one to hold the information of the object's mesh, one to render the surface of the object based on its mesh data, one to represent the volume of the object in terms of physical behaviour and one to represent the object's position and orientation in the scene.

The game components of a game object should be able to access the other components of the object and query them for information (Lambert 2014). In some cases, game components should be able to read and write the values stored in other components of their parent object or of another game object in the scene. Equally, they should be able to query the scene for information regarding other game objects in the scene.

Game objects can also be composed of other game objects hierarchically. In this regard, the topmost game object is called the parent, and all the game objects underneath are called child or children (Unity Technologies 2019).

Furthermore, game objects can be stored as reusable assets. A reusable asset is essentially a template of a game object complete with all its game components, property values and children game objects from which new instances of the game object can be created (Unity Technologies 2019). Reusable assets can be instantiated during scene development or at runtime.

5.1.2 Scripts

As previously mentioned, game components control the behaviour of the game objects to which they are attached. In the context of game engines, a script is a type of game component that consists of a piece of code that implements the developer's own features. Scripts can be used, for example, to trigger game events, modify the properties of other game components or respond to user input (Unity Technologies 2019).

Similar to other game components, a script defines a blueprint for the properties and behaviour of a type of component, but its code is not active unless an instance of the script is attached to a game object in the scene (Unity Technologies 2019). This means that several of the game objects in the scene can have a different instance of the same script attached to them. Each instance executes its code independently, producing different results in each game object.

In the Unity game engine, scripting supports the C# programming language natively. Furthermore, it allows integration with Microsoft's integrated development environment (IDE) Visual Studio, which provides a more sophisticated C# development environment. This feature was leveraged in the implementation of the proposed framework presented in this chapter.

5.1.3 Editor and play modes

During the game development process in Unity, a project is created by loading assets into scenes in "editor" mode. In "play" mode, the project starts and runs. When play mode starts, the scene is loaded, and the scripts in it are reset to their initial values (Unity Technologies 2019). It is in play mode that the game engine executes the code in the scripts; therefore, play mode is also referred to as runtime.

In the implementation of the proposed framework, scripts and reusable assets are developed in editor mode while the simulation runs in play mode. Simulation results are also displayed in play mode, but they can be exported for visualising outside of the game engine.

5.2 Environment

As discussed in Section 4.3, the environment module of the proposed framework consists of three main types of components: Constructable Elements, generic simulation models of construction activities and models of resources. These components can be reused as-is across multiple projects that involve similar construction products or resources. In this regard, they are implemented in the game engine as game components or as reusable assets that can be instantiated either in editor mode or at runtime, depending on their use.

In the remainder of this section, the implementation of each type of component in the game engine is described.

5.2.1 Constructable Elements

The Constructable Element is an interface between a BIM element and its corresponding construction activity simulation model. Its main purpose is to extract and enrich the data from the BIM element and feed it as input to the simulation model.

The Constructable Element is implemented in the game engine as a game component through scripting. Figure 5.1 depicts the base class of the Constructable Element game component. As the figure shows, the Constructable Element game component is attached to a game object, which also has a BIM element attached. The Constructable Element extracts and enriches the data of such BIM element.

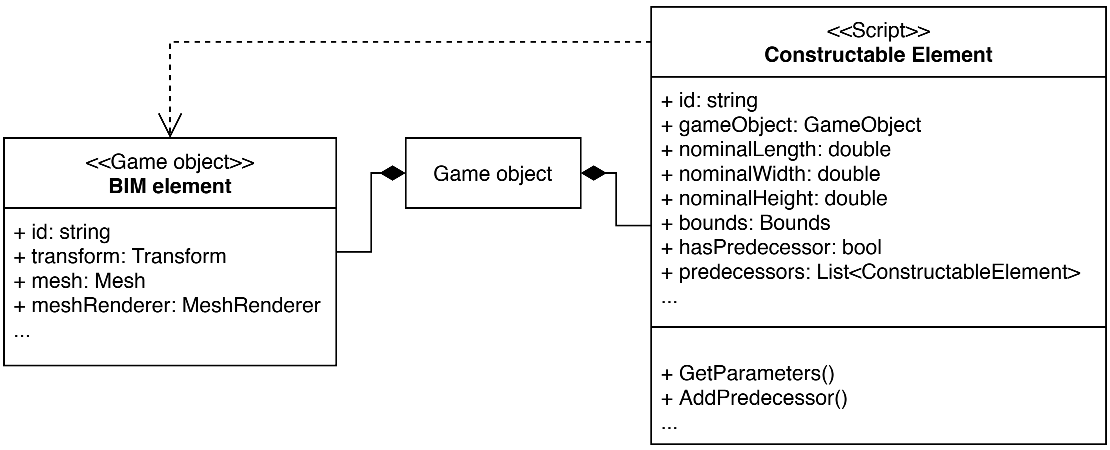


Figure 5.1: Constructable Element class diagram

The Constructable Element class should be extended to extract and enrich BIM data from each type of element in the model. The fields of the Constructable Element should reflect the requirements of its corresponding construction activity simulation model. However, once the Constructable Element for a given type of BIM element has been modelled, it can be instantiated and attached to various game objects that represent that type of building component in the scene. Furthermore, such game component can be reused in different projects with the same type of building components.

Fields and methods

The *id*, *nominalLength*, *nominalWidth*, *nominalHeight* and *bounds* fields of the class are read from the BIM element using the *GetParameters* method. The id property is the unique identifier of the element. The geometric properties or the element can be used to compute the required materials to build it. The bounds of the element are used to find its location and its topological relationship with other elements in the scene.

The *hasPredecessor* field indicates if there are other elements that need to be built before the element in question, while the *predecessors* field indicates which elements have this requirement. The *AddPredecessor* method populates the latter field automatically according to predefined customised rules.

As mentioned earlier, the Constructable Element should be extended to reflect the requirements of its corresponding construction activity simulation model. This extension should include the appropriate methods to compute the required materials to build the element, the rules to establish interdependencies and the additional parameter-based information that the simulation model requires. For example, Listing 5.1 depicts a method that establishes a predecessor interdependency between a Constructable Element and the building components supported by it based on their topological relationship. This method belongs to a class that extends the Constructable Element, but is not part of the base class as it may not be relevant to all the BIM elements in the model.

```
void SupportsAnotherElement()
{
    Bounds bounds = gameObject.GetComponent<Renderer>().bounds;
    float y = bounds.center.y + bounds.extents.y;
    float xmin = bounds.center.x - bounds.extents.x;
    float xmax = bounds.center.x + bounds.extents.x;
    float zmin = bounds.center.z - bounds.extents.z;
    float zmax = bounds.center.z + bounds.extents.z;
    // Points (xmin,y,zmin), (xmax,y,zmin), (xmin,y,zmax) and
        (xmax,y,zmax) represent the top surface of this constructable
        element

    ConstructableElement[] all =
        FindObjectsOfType<ConstructableElement>();

    foreach(ConstructableElement element in all)
    {
        Vector3 bottom = new Vector3(
            element.bounds.center.x,
            element.bounds.center.y - element.bounds.extents.y,
            element.bounds.center.z);

        if (bottom.y == y &&
            bottom.x < xmax && bottom.x > xmin &&
            bottom.z < zmax && bottom.z > zmin)
        {
            element.AddPredecessor(
                gameObject.GetComponent<ConstructableElement>());
        }
    }
}
```

Listing 5.1: A method that establishes a predecessor interdependency between a constructable element and the building components supported by it

Instantiation

Game components of the Constructable Element class are instantiated and attached to their corresponding game objects during the pre-processing phase of the proposed framework. This process can be done either automatically at runtime, or manually in editor mode, as detailed in Section 5.4.

5.2.2 Generic simulation models of construction activities

The generic simulation models represent the different construction activities required to build the BIM elements of the input BIM model. These models are generic until their input parameters are fed by a Constructable Element that extracts and enriches the data from their corresponding BIM element.

Generic simulation models are implemented in the game engine as game components through scripting. Similar to Constructable Elements, these game components are attached to a game object, which also has a Constructable Element game component attached. Figure 5.2 depicts the base class of the generic simulation model game component.

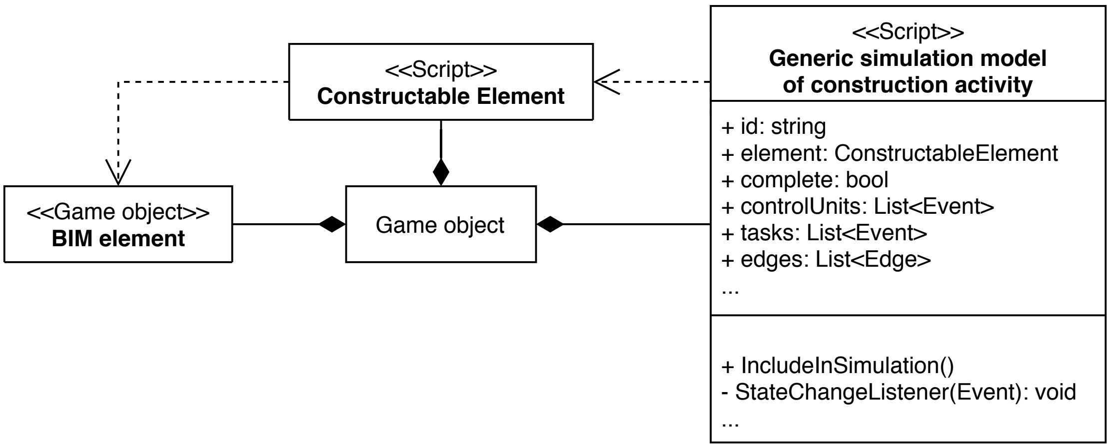


Figure 5.2: Generic model of construction activity class diagram

The generic model of construction activity class should be extended for each type of element in the model. Nevertheless, the developed game components can be instantiated and attached to multiple game objects that represent the products of the construction activity that they represent.

Model development

Each game component of the generic simulation of construction activity type consists of a script developed in the C# programming language using the SharpSim library, an open-source discrete-event simulation (DES) code library developed by Ceylan and Gunal (2011). This library was chosen because a review of state-of-the-art open-source DES software used for decision support in operations research revealed that SharpSim was the only tool written in the C# programming language (Dagkakis and Heavey 2016), which is supported by Unity.

Models developed using the SharpSim library are composed of events and edges. In SharpSim, an event causes a state change in the modelled system, while edges establish the relationships between the events and the flow of the system. Each event in the model needs a “State Change Listener”, represented by a private method that takes the event as its only argument. State change listeners are related to C# event handling mechanism and help connect SharpSim events with C# events (Ceylan and Gunal 2011). Basically, each generic simulation model consists of events and edges that encapsulate the logical relationships between the tasks that comprise the modelled construction activity, and state change listeners that contain the code that produces and tracks the changes in the system when events occur. Finally, SharpSim provides an abstract *Entity* class, which can be extended to create model related entities (Varol et al. 2011).

Figure 5.3 illustrates the class diagram of a basic SharpSim simulation model. These classes are adopted in this implementation of the proposed framework to integrate construction simulation and BIM as described in the following paragraphs.

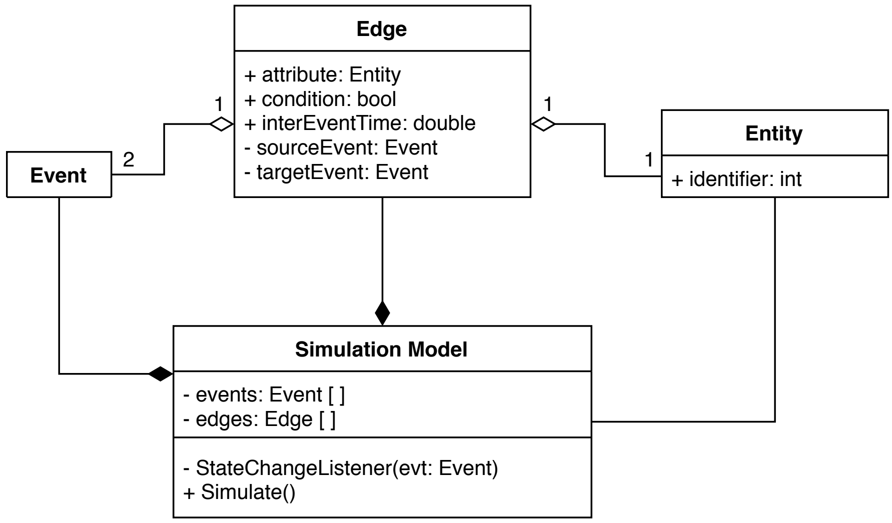


Figure 5.3: SharpSim model class diagram

Fields

As detailed in Section 4.3.2, generic simulation models of construction activities are composed of simple tasks, control units and parametric requirements. As shown in Figure 5.2, both the tasks and control units are implemented as SharpSim *Events*.

The *edges* field contains the SharpSim *Edges* that connect the events of the model. The *condition* field of the SharpSim edges determines whether the flow of entities through the edge is allowed or not. In SharpSim models, the simulation clock is advanced by edges via their *interEventTime* field. Thus, as shown in Figure 5.4, edges that connect control units to tasks do not have a value for the interEvent-Time field, while those that connect tasks to control units take the value of the duration of the task.

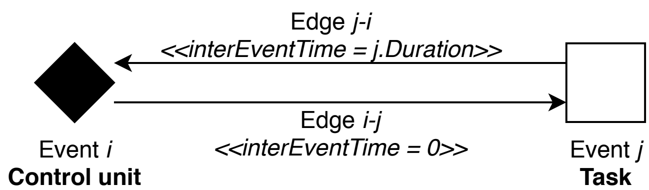


Figure 5.4: Task duration as interEventTime of a SharpSim Edge

As detailed in Section 4.3.2, the duration of the tasks in the proposed framework can be sampled from a probability distribution to represent the stochastic nature of construction tasks. SharpSim implements this feature natively (Ceylan and Gunal 2011), however, its built-in library of distributions is limited and the method to input different distribution parameters is not straightforward. To overcome these issues, tasks duration in the generic simulation model class in this implementation, were modelled using the Math.Net Numerics library, which provides methods and algorithms for numerical computations and it is written in C# (Ruegg et al. 2019).

The *element* field of the generic simulation model game component is the constructable element game component, which, as previously discussed, contains the parametric requirements of the simulation model. Finally, the boolean *complete* field indicates whether the construction activity is complete or not.

Methods

As previously stated, a state change listener method is required for each SharpSim event in the model. This means that the generic simulation model class has one of these methods for every control unit and task modelled. These methods contain the code that produces and tracks the changes in the system when events occur. State change listeners of control units determine the flow of the system by changing the value of the condition field of the edges that connect them to tasks or to other control units. On the other hand, state change listeners of tasks produce changes in the resources of the system.

The *IncludeInSimulation* method is used to instantiate the events and edges of the simulation model of the construction activity, and to add them to the list of events and edges of the simulation model of the simulation project, as detailed in Section 5.4. This method includes the instantiation of an edge between the topmost control unit of the construction activity simulation model and a control unit in a higher level of the control structure hierarchy. By creating this edge, the construction activity simulation model becomes a submodel of the distributed simulation composed of all the simulation models of construction activities in the scene.

Instantiation

Similar to Constructable Element game components, game components of this class are also instantiated and attached to their corresponding game objects during the preprocessing phase of the proposed framework. This process can be done either automatically at runtime, or manually in editor mode, as detailed in Section 5.4.

5.2.3 Models of resources

As detailed in Section 4.3.3, resources in the proposed framework are classified in three categories: materials, workers and equipment. Each of these categories extends the resource class following an object-oriented modelling approach, and each resource instance consists of the combination of two components: a 3D model and a set of attributes that model the resource in the simulation model. In the game engine, each of these components is implemented as a game component attached to the same game object, as depicted in Figure 5.5. The resulting game object composes a reusable asset, which, as discussed, can be instantiated during scene development or at runtime.

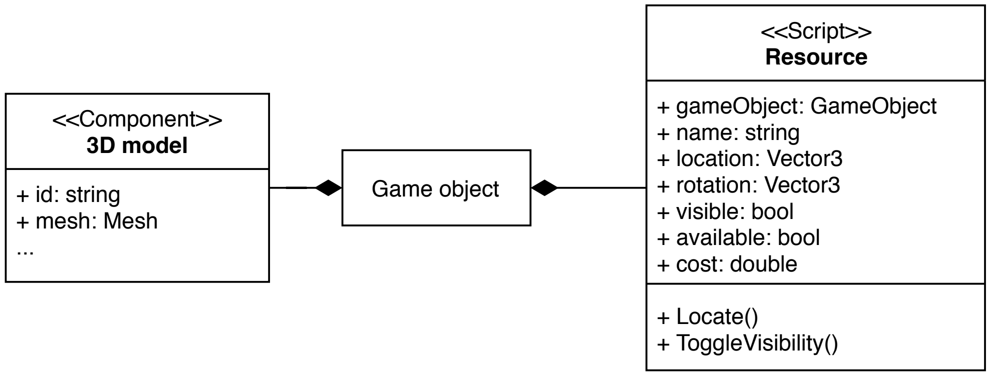


Figure 5.5: Resource game object

Instantiation

Resources of the worker and equipment type are instantiated based on user input. On the other hand, materials are instantiated at runtime during simulation.

5.3 User input

As introduced in Section 4.4, the proposed framework enables users to develop a construction simulation model based on an existing BIM model. Furthermore, it allows users to run the simulation under different resource availability and project-specific constraints scenarios. This section details some technical aspects regarding user input in the proposed framework.

5.3.1 BIM model

In the proposed framework, users are able to import their own BIM models into the game engine. Once imported into the scene, each BIM element of the model becomes an individual game object with the necessary game components to provide the system with the relevant data to instantiate a simulation model of the construction activity required to build the BIM element. Such game components include first, a transform component, which holds information regarding the position, orientation and scale of the BIM element relative to a known reference. Second, a mesh component, which is a collection of triangular boundaries, which collectively forms the 3D shape of the element. Third, a renderer component, which renders the element in the scene based on its mesh and material properties. Figure 5.6 depicts the class diagram of a BIM element as a game object.

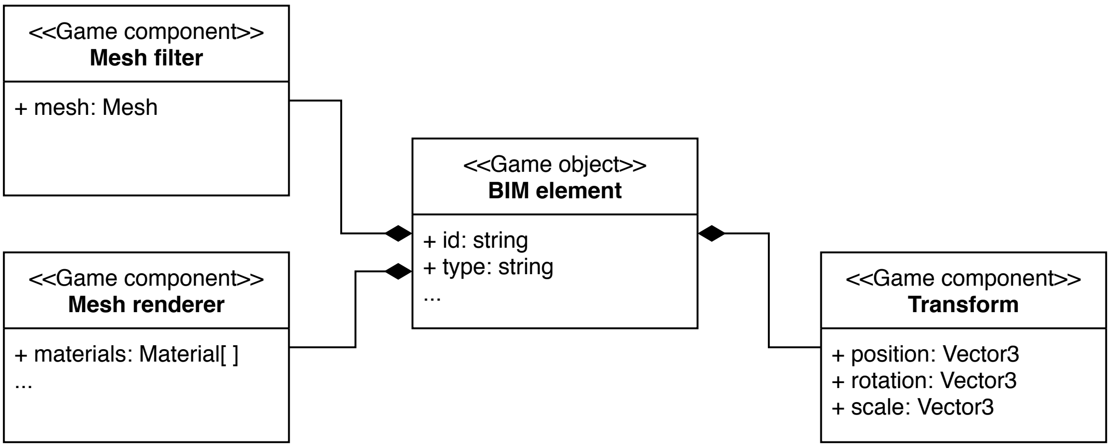


Figure 5.6: BIM element as game object class diagram

Importing the BIM model into the game engine can be achieved using two different methods, namely, importing the model as a filmbox (fbx) file or as an Industry Foundation Classes (IFC) data model. As discussed in Chapter 4, the rationale behind this feature is to enable interoperability with non-IFC-based 3D modelling tools, and to provide flexibility in the 3D modelling approach of designers. The advantages and drawbacks of both methods are discussed below.

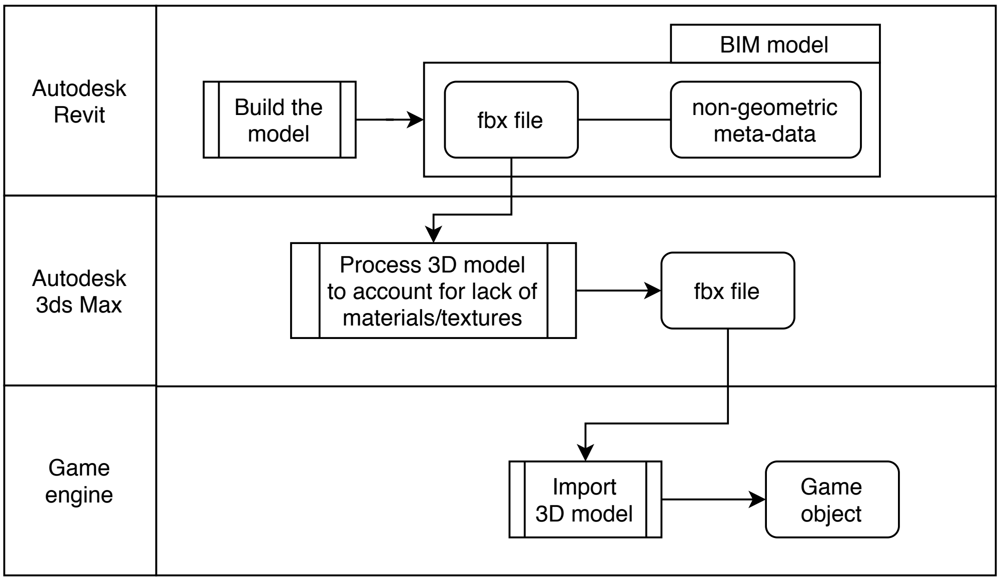
Importing an fbx file into the game engine

Unity natively supports filmbox (fbx) files, which can be imported from many 3D modelling applications (Unity Technologies 2019). In this regard, Broadbent (2011) and Dalton and Parfitt (2013) proposed a workflow to import BIM models into Unity as fbx files. The workflow has been adopted by several researchers working with BIM and game engines (Bille et al. 2014; W. Wu and Kaushik 2015; Natephra et al. 2017; Ben-Alon and Sacks 2017). This method consists in exporting models produced in Revit to an fbx file, processing the fbx in Autodesk 3ds Max and importing it into the game engine. The additional processing of the fbx file in Autodesk 3ds Max was suggested to prevent the loss of colours and detailed textures that occurs when the model is imported directly into the game engine. This workflow is depicted in Figure 5.7a.

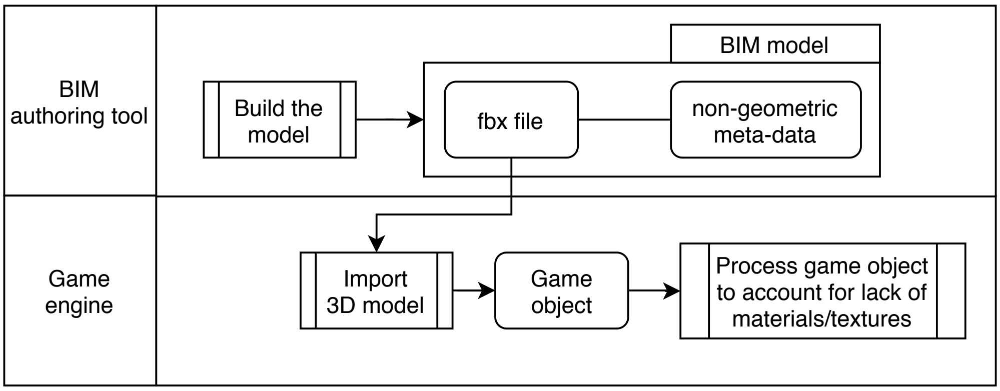
A similar workflow was adopted in the proposed framework. However, the processing of the fbx file in Autodesk 3ds Max was omitted. The reason for this omission is that most game engines, including Unity, allow users to import or create their own materials and textures, and adjust their properties to achieve the desired visualisation without the need of another piece of software. This approach is shown in Figure 5.7b.

A drawback of both approaches shown in Figure 5.7 is that the non-geometrical meta-data of the model is lost when the fbx file is imported into the game engine. Figure 5.8 illustrates a solution to this issue. Non-geometric meta-data can be assigned back to the model elements that require it within the game engine by attaching a game component that contains the desired data to the relevant game object through scripting.

An advantage of importing BIM models as fbx files into the game engine is that the model can be modified in the editor mode of the game engine. This means that additional functionality can be added to each BIM element within the environment before going into play mode. Assigning non-geometric meta-data by attaching scripts to game objects or changing materials and textures of the different game objects are examples of this added functionality.



(a) Option 1: Additional processing in Autodesk 3ds Max



(b) Option 2: Additional processing in game engine

Figure 5.7: Options to import an fbx file into a game engine

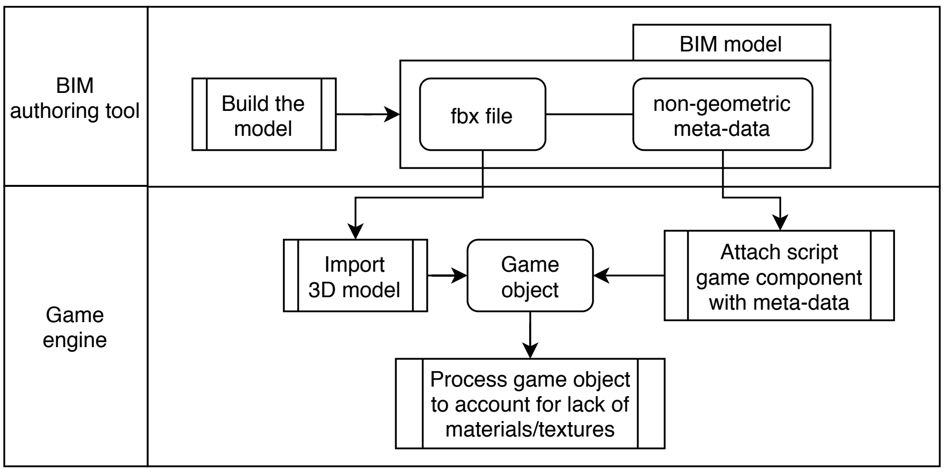


Figure 5.8: BIM meta-data as script game component

Importing an IFC data model into the game engine

At the time of writing, commercial game engines do not support direct import of Industry Foundation Classes (IFC) files. The extensible building information modelling (xBIM) toolkit, developed by Lockley et al. (2017), was tested during the implementation phase of the proposed framework. This toolkit was chosen among the few alternatives available to import IFC into game engines because it is free, open-source, and it has up-to-date activity in its community forum.

Using the xBIM toolkit, the geometry data contained in an IFC file is read and saved into meshes. A new game object is created in the game engine for each mesh in the model. Through scripting, the non-geometric meta-data contained in the IFC file can be attached as a game component to each game object. This meta-data includes information regarding the materials and textures, which is assigned to a renderer game component that is also attached to the game object, eliminating the need to process the model to improve visualisation of the objects. Furthermore, additional type-based game components can also be attached through scripting to the game objects that are created from the IFC model. These game components can enhance the functionality of the model within the game engine, for example, triggers to open doors upon approaching them during model navigation. Figure 5.9 depicts this process.

An inconvenience of xBIM compared to more expensive alternatives is that it allows to import IFC files into the game engine only in play mode, which does not allow for further modifications of the model in editor mode before starting the application. However, it enables the automation of model preparation upon importing the IFC data model.

Figure 5.10 shows an IFC file imported into a game engine using the xBIM toolkit.

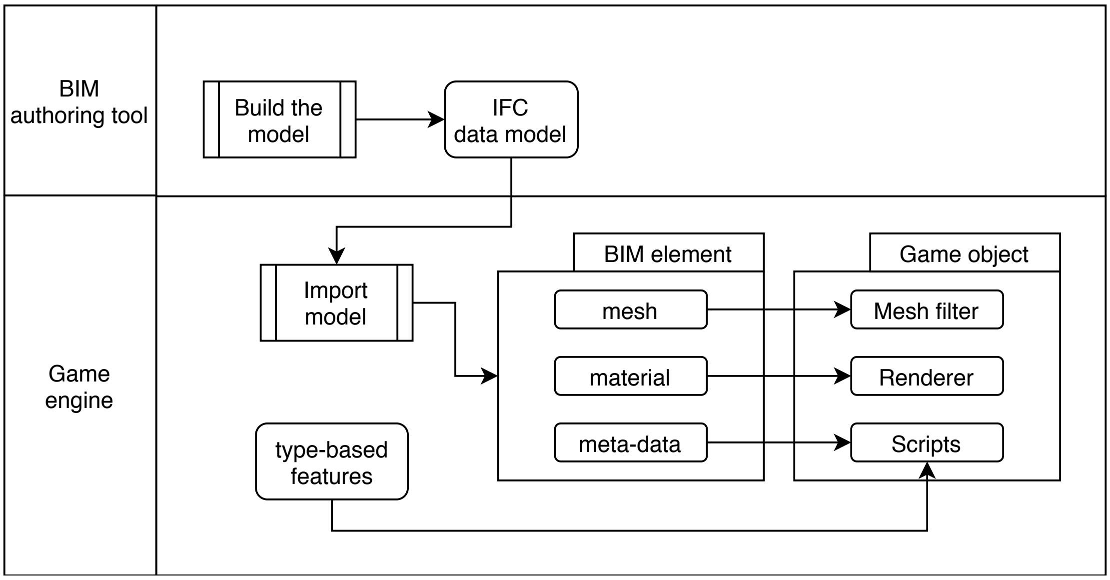


Figure 5.9: Workflow to import an IFC data model into a game engine

A script that displays the BIM properties of an element when it is selected was attached to every beam element in the model as an example of type-based functionality.

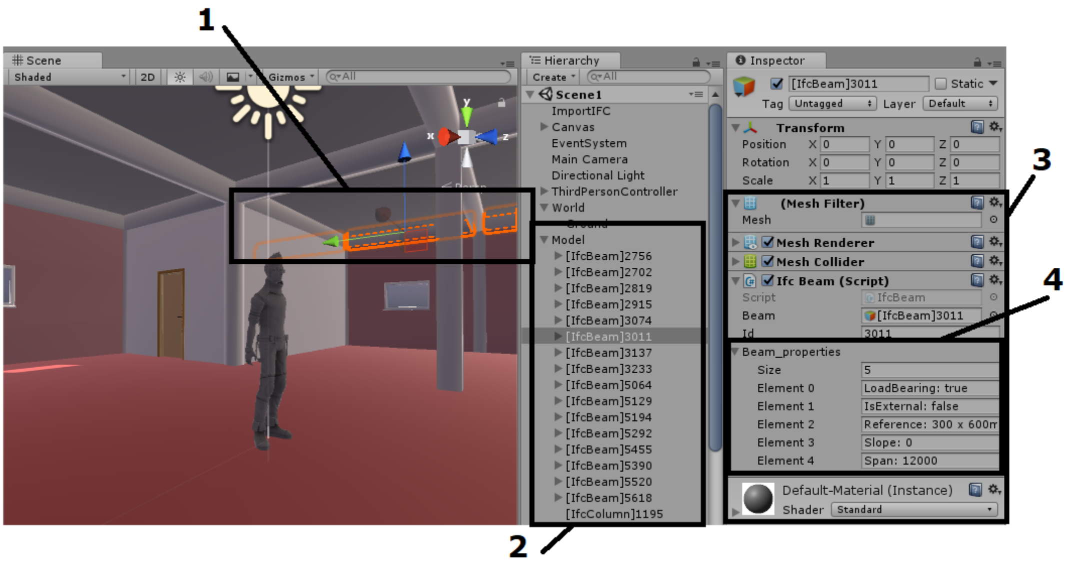
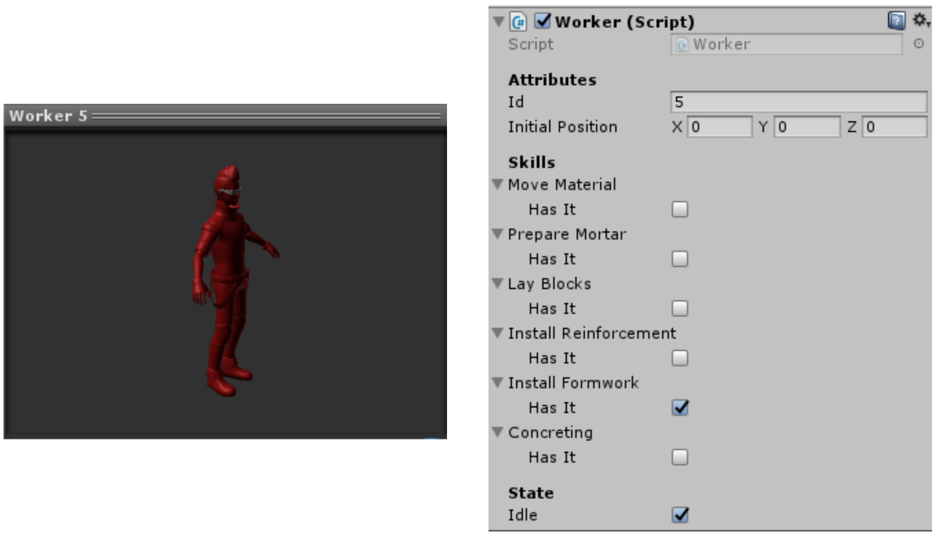


Figure 5.10: IFC file imported into a game engine. 1) Selected element in the Scene view 2) Imported model hierarchy 3) Components added to the game object 4) BIM properties extracted from the IFC file

One of the advantages of importing a BIM model into a game engine using the IFC format is that the model's non-geometrical information is not lost as it happens when the model is imported as an fbx file. On the other hand, the drawbacks of this workflow are that it is not as straight forward as importing an fbx file, and that it requires additional tools that may not be easily acquired or compatible with all commercial game engines

5.3.2 Resource-related input

As discussed in Section 4.4.2, users are able to define some resource-related parameters to simulate different scenarios. During scene development, users can instantiate the desired number of resources of the worker and equipment types by dragging and dropping the corresponding reusable assets into the scene in the user interface of the game engine. The attributes of each instance can be defined in the inspector window of the user interface. Figure 5.11 illustrates an instance of the worker type in the inspector window of the game engine. The 3D model of the resource can be visualised in Figure 5.11a, and its properties can be defined in editor mode as shown in Figure 5.11b.



(a) Worker 3D model (from Unity Technologies (2017))

(b) Worker script game component

Figure 5.11: Resource of the worker type in the inspector window

Storage sites of each type of raw material are implemented as game objects with a transform that determines its location, and a simple script game component that determines its capacity. Users can define both the location and capacity of each storage site in the inspector window of the relevant game object.

Equally, users can input compatible independent models to represent other project-specific constraints by attaching such models to active game objects in the scene as script game components. Such game objects can be "empty" game objects, meaning that they act solely as a container for those game component (i.e. they do not need to have a mesh or a physical representation in the scene).

5.4 Pre-processing module

As described in Section 4.5, in this module of the proposed framework, a construction simulation model is semi-automatically generated based on the input BIM model.

In the game engine, this process is implemented within a script that first attaches two game components, a Constructable Element and a generic simulation model of a construction activity, to each BIM element game object in the scene based on its type. Then, it executes the GetParameters method of each Constructable Element game component, which extracts and enriches the required data from its corresponding BIM element, and feeds it into the generic simulation model game component. Finally, it executes the IncludeInSimulation method of the now specific simulation model of the construction activity of each BIM element, which adds its events and edges to the simulation of the construction project. Listing 5.2 shows the process described above.

Once this process is complete, game objects that represent BIM elements in the

```
public void Preprocessing()
{
    BIMElement[] elements;
    ConstructableElement constructable;
    ActivitySimulationModel simulationModel;

    elements = FindObjectsOfType<BIMElement>();
    foreach (BIMElement element in elements)
    {
        //These methods attach appropriate game components based on the element's type
        AttachConstructableElement(element);
        AttachActivitySimulationModel(element);

        constructable = element.GetComponent<ConstructableElement>();
        constructable.GetParameters();

        simulationModel = element.GetComponent<ActivitySimulationModel>();
        simulationModel.IncludeInSimulation();
    }
}
```

Listing 5.2: Preprocessing script

scene have the necessary game components to start the simulation. Figure 5.12 depicts two game objects created from two BIM elements. The properties of the Constructable Element game component of the game object highlighted in orange are shown in the inspector window highlighted in red. The figure shows that the Constructable Element script has extracted data from the BIM element, and populated its fields. Moreover, it has enriched the data by establishing a predecessor interdependency with the game object that supports it.

Furthermore, the simulation models of construction activities in the scene are no longer generic since they represent the construction activities required to build the BIM elements associated with them. Figure 5.13 illustrates how a generic simulation model becomes specific to a BIM element after taking its parameters, and how the model of the activity is aggregated to the simulation model of the construction project.

Figure 5.14 depicts a sequence diagram regarding how the distributed simulation

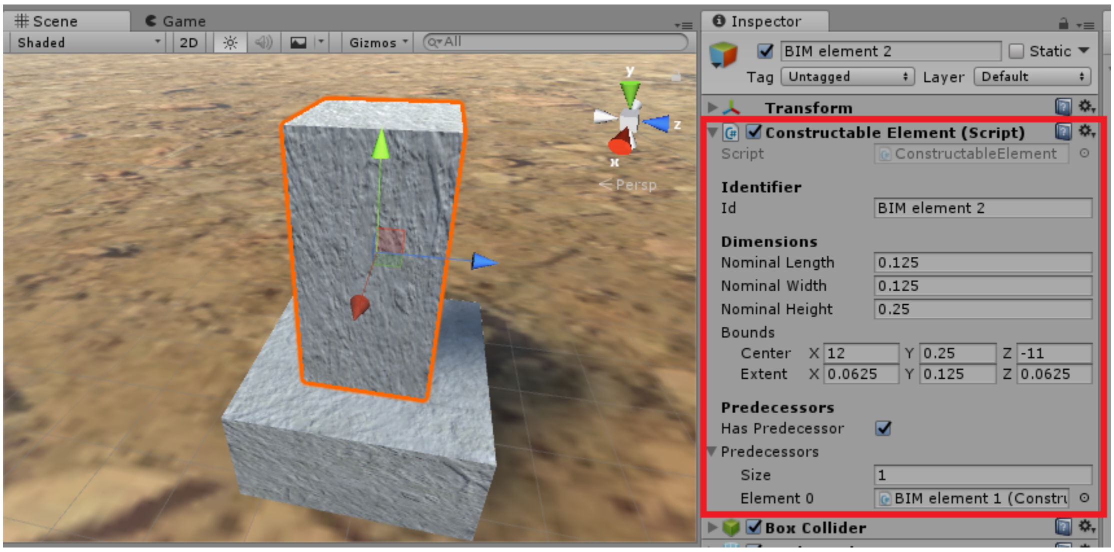


Figure 5.12: Game objects with Constructable Element scripts

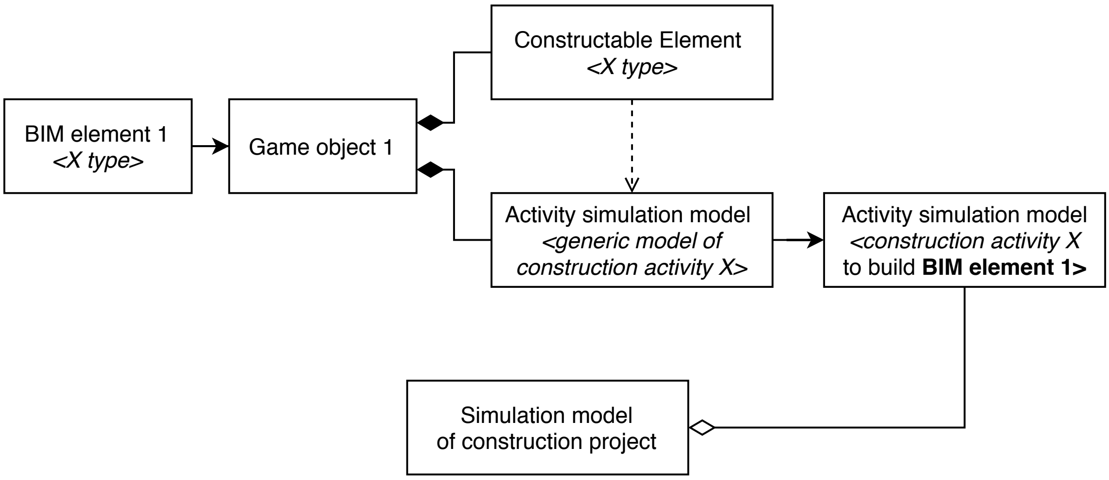


Figure 5.13: Aggregation of simulation model of construction activity to simulation model of construction project

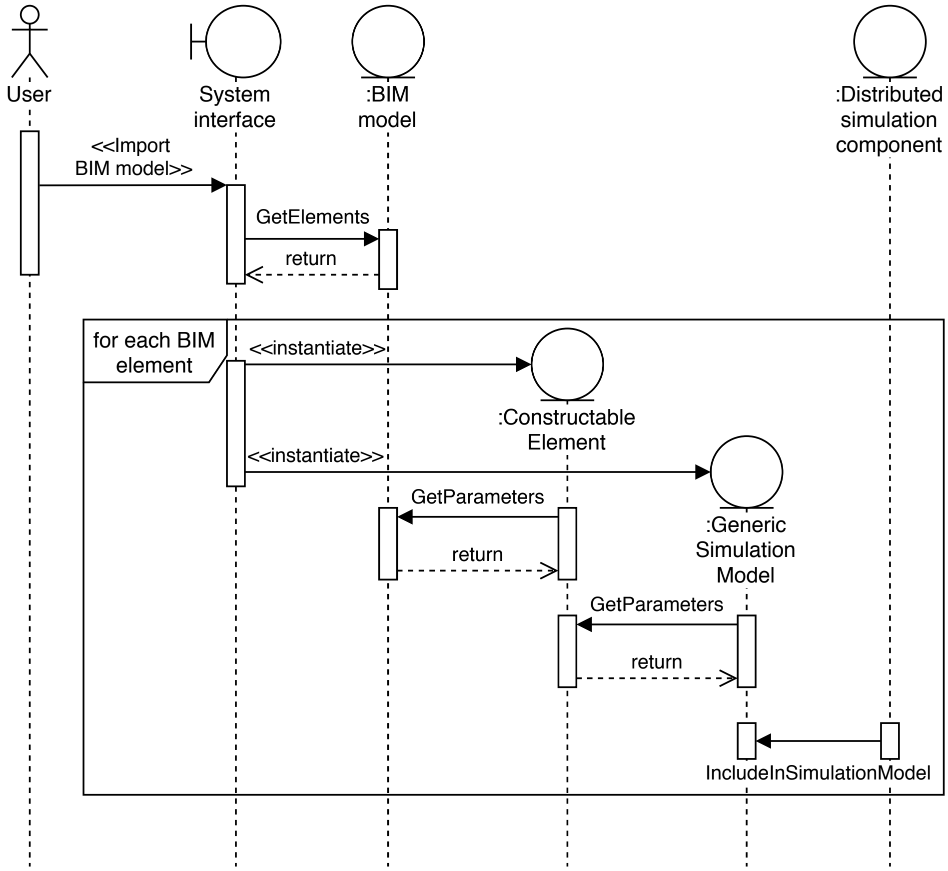


Figure 5.14: Sequence diagram showing how federates are included in the federation

component, implemented as a C# script, includes the now specific construction activity federates in the construction project federation.

5.5 Simulation module

As described in Section 4.6, the simulation module of the proposed framework is composed of six components: a distributed simulation, a construction project simulation model, resources simulation models, modifying parameters, other simulation models and trace files. This section describes the implementation of each component of this module.

5.5.1 Distributed simulation

The distributed simulation component of the simulation module is implemented as a script game component developed in C# using the SharpSim library (Ceylan and Gunal 2011). Figure 5.15 depicts the class diagram of the distributed simulation component of this module and its relationships with other components.

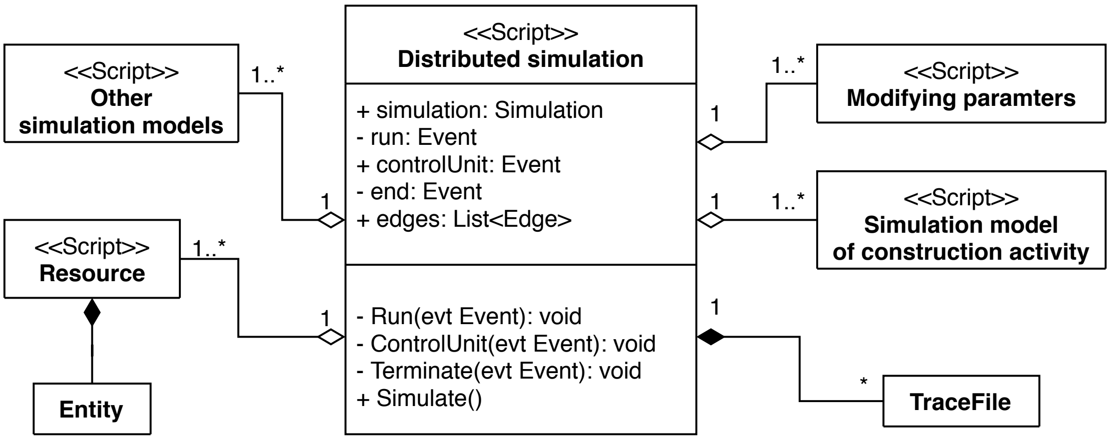


Figure 5.15: Class diagram of the distributed simulation component

As the figure shows, the distributed simulation component is composed of three SharpSim events. First, the run event, which determines the beginning of the simulation. Second, a control unit event, which is the highest control unit in the control structure of the project. If needed, further control units can be instantiated at this point. Third, an end event, which determines the ending of the simulation. As detailed in Section 5.2.2, each SharpSim event requires a state change listener method. In addition, it has an instance of the SharpSim *Simulation* class. The method *Simulate* instantiates and executes the simulation object, which includes all the events and edges in the individual submodels of construction activities and other simulation models.

5.5.2 Construction project simulation model

As discussed before, this component is composed of the specific simulation models of construction activities, which are instantiated during the preprocessing phase of the proposed framework, as detailed in Section 5.4.

5.5.3 Resources simulation models

During the simulation, the distributed simulation component verifies if a task's required resources are available by scanning the scene to find resources that match the task's requirements. The implementation of resources was discussed in more detail in Section 5.2.3.

5.5.4 Modifying parameters

As discussed in Section 4.6.4, modifying parameters are either direct user input or based on features extracted from the user input BIM model. How the latter affect the simulation model is discussed in Section 5.4, since this occurs during the preprocessing phase of the proposed framework. The parameters directly input by the user are discussed in Section 5.3.2. Before the simulation starts, the distributed simulation component updates the parameter-based rules in its control units based on user input parameters through scripting.

5.5.5 Other simulation models

Based on the distributed simulation approach adopted in this research, other simulation models can run concurrently with the construction simulation model. These models can be implemented as script game components attached to active "empty" game objects in the scene. In order to be compatible with the implementation presented in this chapter, such models should be developed with SharpSim components. However, the proposed framework can be implemented to support other simulation modelling tools due to its generic conceptualisation.

5.5.6 Trace files

Trace files are implemented following an object-oriented approach. Figure 5.16 shows the class diagram of the trace file class in this implementation of the proposed framework. A new object of the trace file class is instantiated whenever there is a change in the modelled system. The instantiation is invoked in the state change listener methods of the events that produce the changes in the system. The trace file object contains all the relevant information to produce the reports in the visualisation module.

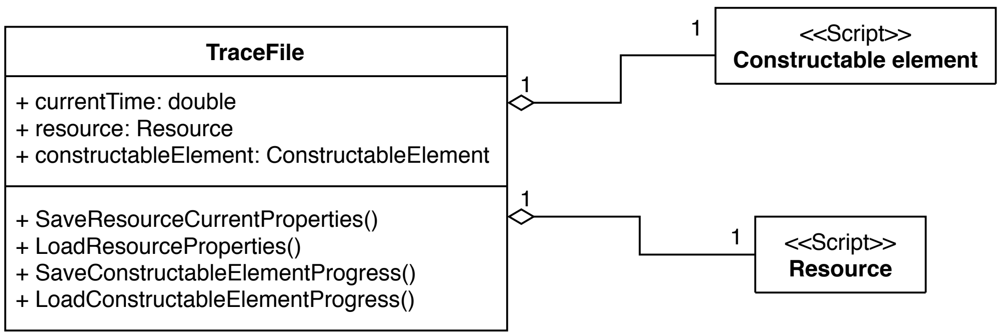


Figure 5.16: Class diagram of the trace file implementation

5.6 Visualisation module

As described in Section 4.7, the visualisation module of the proposed framework utilises the trace files generated in the simulation module to generate reports and animations to support planning and decision-making. The implementation of this module leverages the capabilities of the game engine as described in the remainder of this section.

5.6.1 Reports

A user interface (UI) was developed using the Unity UI toolkit, a game object-based UI system that uses components and the game view to arrange, position, and style user interfaces (Unity Technologies 2019). Using this toolkit, scripts to plot the relevant results in the corresponding reports were developed in the C# programming language. Figure 5.17 depicts the developed user interface.

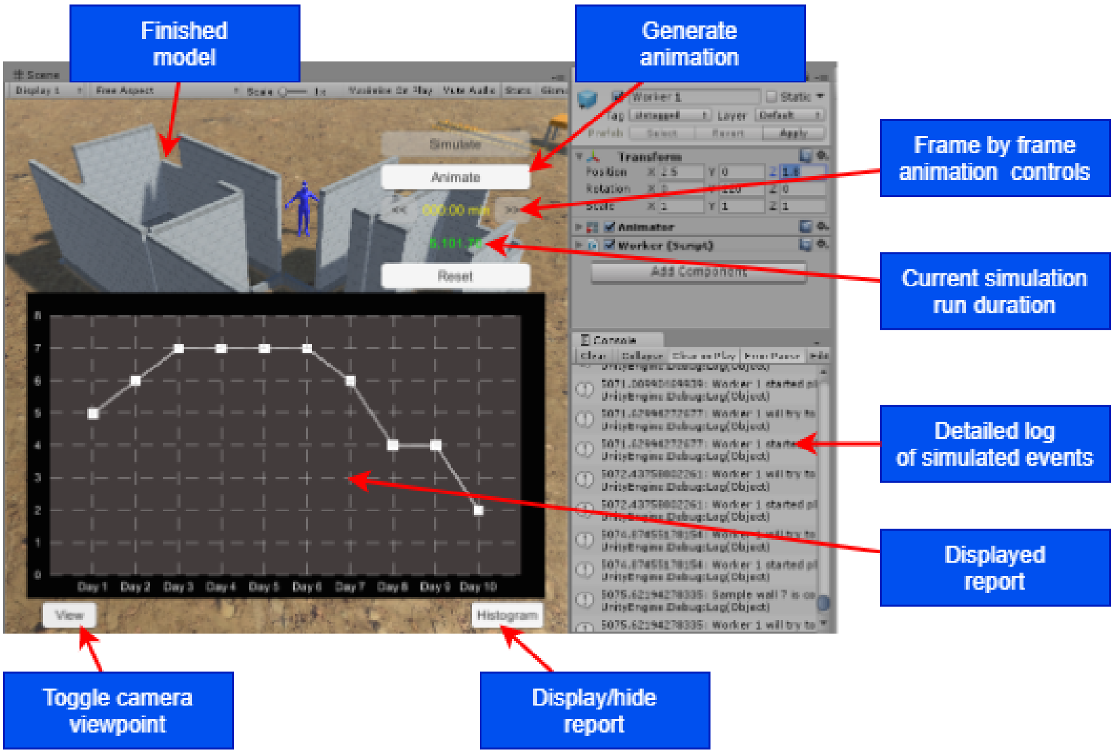


Figure 5.17: Visualisation module user interface

5.6.2 Animations

Users can generate a simulation-based animation after the simulation is complete using the *Animate* button shown in Figure 5.17. Such an animation can be visualised continuously or frame by frame. As previously discussed, even when visualised continuously, the animation only shows the positions of resources at the beginning and ending of the tasks in which they are taking part. Visualisation of the continuous motion of resources has not been implemented at the moment of writing.

5.7 Summary

This chapter described the implementation of the proposed framework for the integration of construction simulation and building information modelling within the Unity game engine. It also discussed the game engine capabilities leveraged in the proposed framework, and how they were used to model its components and achieve the sought functionality. The chapter presented the classes developed to implement the components of the proposed framework introduced in Chapter 4 using the Unified Modelling Language. The next chapter further illustrates the implementation and usage of the proposed framework by presenting a case study based on a typical masonry construction problem that follow the implementation approach introduced in this chapter.

Chapter 6

An illustrative case study to assess the feasibility of the implemented framework

The previous chapter described the implementation of the proposed framework for the integration of construction simulation and building information modelling within the Unity game engine. It described the game engine concepts appropriated in this research, and how they were used to achieve the intended functionality of the framework.

This chapter presents a case study to evaluate the employment of the implemented framework. The case study is based on a typical masonry construction problem, and it showcases the usefulness of the proposed framework as a planning tool. The chapter analyses and discusses the results obtained from the case study.

6.1 Objectives

The main objective of this case study is to evaluate the applicability of the proposed framework as a planning tool. A case study is considered an appropriate approach for such an assessment (Kitchenham et al. 1995). Additionally, this case study also seeks to illustrate the development of the required components of the framework, which should guide users on how to design customised environments based on the proposed framework. The example also demonstrates and assesses scenario development for what-if analysis. Finally, this case study explains how to utilise the obtained results for informed decision-making.

6.2 Methodology

The case study presented in this chapter was performed according to the following methodology:

- A typical construction problem was selected to test the suitability of the framework.
- Model assumptions were documented.
- The studied construction activity was broken down into several simple tasks.
- A conceptual model of the system was developed and documented.
- Based on the conceptual model, a generic simulation model of the selected construction activity was developed following the implementation approach presented in Section 5.2.2.
- An instance of the Constructable Element class was developed for the selected construction activity.
- A BIM model was imported into the scene following the steps described in Section 5.3.1.
- A simulation model based on the imported BIM was developed using the proposed framework.

- The simulation model was validated using a combination of two techniques, namely, animation and traces.
- 14 different scenarios were developed to represent solutions to the problem selected for the case study. Three simulation outcomes were selected to provide insights to support the decision-making process in the selected problem.
- Each modelled scenario was simulated twenty times and the results were compared to provide recommendations. An empirical cumulative distribution function (ECDF) was plotted based on the obtained results. The ECDF was compared to its corresponding normal cumulative distribution function (CDF) to verify that the number of simulation runs was enough to produce a normal distribution. Additionally, a Kolmogorov-Smirnov goodness-of-fit test was also applied to the results.

6.3 Background

Masonry walls with concrete blocks represents one of the most common construction activities in the industry. Although in the 20th century masonry was displaced for many applications by steel and concrete, it remains of great importance for walls in low- and medium-rise buildings and for internal walls and cladding of buildings where other materials meet the structural function (Hendry 2001).

Furthermore, the construction of masonry walls with concrete blocks is one of the most representative construction activities in terms of the impact of its direct cost on the total cost of the projects that include it. According to a comparison of the cost estimates of several housing projects in South East Mexico, the direct cost of this construction activity represented on average 8.28% of the total cost of each project, surpassed only by rubble stone masonry foundations and pre-cast beam and block slab roofing. The direct costs of the latter two activities had, on average, a higher impact on the total cost of the projects that included them. However, not all the analysed projects included them, since other roofing and foundation systems were employed instead (Osorio-Sandoval 2015). For these reasons, masonry walls with concrete blocks was selected to demonstrate and evaluate the employment of the proposed framework.

A typical masonry crew consists of two types of workers, skilled workers, or masons, and unskilled labourers, or helpers. Depending on the job, the size and composition of this crew may vary. In general, an objective in construction projects is to decrease the crew size to the minimum possible for each job. Contractors usually start the job with a pilot minimum crew and increase it gradually to keep a continuous workflow (Hassanein and Melin 1997).

The presented case study analyses several compositions of crews performing the construction of masonry walls with concrete blocks in a large housing project to assess the employment of the proposed framework.

6.4 Project description

The selected project consists of the construction of 746 one-storey houses as part of a larger housing project in South East Mexico. The case study seeks to determine the number of helpers that a mason needs to keep a continuous workflow based on the distance between the house that the crew is working on and the materials storage site. For this purpose, the site has been divided into four zones, as depicted in Figure 6.1.

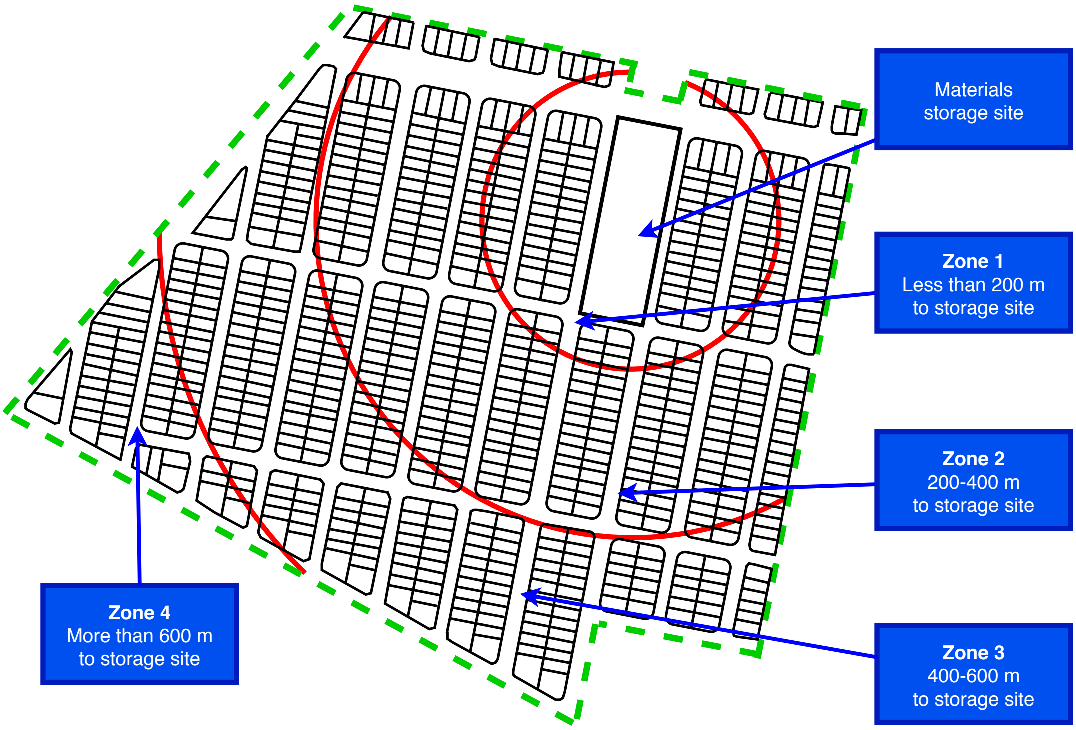


Figure 6.1: Case study site layout

6.5 Model assumptions

This section discusses the rationale behind the assumptions made during the development of the simulation model.

6.5.1 Workers

As stated before, a standard masonry crew consists of two types of workers, masons and helpers. Although in construction sites there may not be a clear demarcation between the tasks that each type of worker performs (Gerek et al. 2015), in this model, it is assumed that masons prepare the mortar and lay the concrete blocks, while helpers move materials from the storage site to where they are needed. Therefore, the minimum masonry crew that can be formed would consist of one mason and one helper.

6.5.2 Materials

There are two materials involved in the construction of masonry walls, namely, the concrete masonry unit (CMU) and the cement mortar. Although there are several types of concrete blocks available in the market, the most commonly used in the construction of masonry walls in the industry in South East Mexico is the 6-inches hollow concrete block with two cores (15x20x40 cm). For this reason, the CMU in this model is this type of block. The cement mortar typically used in this construction activity has a proportion of portland cement and fine aggregate of 1:4 and water. In this model, it is assumed that 0.0008 m^3 of mortar are consumed to lay each CMU.

6.5.3 Equipment

In this model, it is assumed that the usual tools required in the construction of masonry walls, such as trowels and chisels, are available at all times. On the other hand, material handling and processing pieces of equipment are modelled as independent resources based on the following assumptions: 1) Wheelbarrows are used to carry concrete blocks, fine aggregate and cement. It is assumed that wheelbarrows have a capacity to carry eight concrete blocks, 0.09 m^3 of fine aggregate or a 50 Kg cement bag at a time. 2) Buckets are used to carry water and mortar. It is assumed that buckets have a capacity of 0.02 m^3. 3) A concrete mixer is used to prepare the mortar. It is assumed that the concrete mixer produces 0.14 m^3 of mortar with one bag of cement, 0.16 m^3 of fine aggregate and two buckets of water.

6.5.4 Related activities

Tasks and resource related to scaffolding are not considered part of the modelled system, although constraints on scaffolding resources can affect the labour productivity, duration and cost of the construction of masonry walls with concrete blocks. There are several project-specific variables that have an impact on scaffolding activities, including the scaffolding's type, size, movement, assembly and disassembly patterns (Kim et al. 2018). These variables need to be defined in advance as part of the strategy for temporary structures (Jin et al. 2020). Moreover, the construction of different scaffolding types is subject to varied labour productivity and resource constraints (Hou et al. 2017). For these reasons, it is considered that scaffolding activities are independent of the construction of masonry walls with concrete blocks. Using the proposed framework, however, a dependency between the installation of scaffolding and concrete block laying at a predefined height could be established, for example, to study the interaction between these linked activities.

In the modelled system, the reinforced concrete plinth bases of the walls are assumed to be already built.

6.5.5 Interruptions

Interruptions due to external factors, such as weather conditions, labour absenteeism or material delays were not considered in the modelled system.

6.6 Conceptual model

This section documents the basis upon which the generic simulation model of the selected construction activity was developed considering the assumptions previously outlined. A work breakdown structure (WBS) was constructed first in order to identify the simple tasks that compose the activity. Then, a conceptual model was developed.

6.6.1 Simple tasks

The selected construction activity was first broken down into simpler and more manageable tasks considering resource allocation factors. For example, moving the concrete blocks from the storage site to the wall construction site was broken down into three separate simpler tasks performed in sequential order, namely, loading the blocks into an empty wheelbarrow at the blocks' storage location, moving the wheelbarrow loaded with the blocks from the storage location to the wall construction site, and unloading the blocks near the wall construction site. In this case, the first and third tasks are not affected by the location of the storage site of the concrete blocks. However, the duration of the second task depends on the distance between the storage and the wall construction sites. Table 6.1 shows the WBS for the modelled activity.

6.6.2 System behaviour

Figure 6.2 illustrates the conceptual model of the construction of a masonry wall with concrete blocks based on the WBS and the assumptions outlined earlier. In this figure, the squares represent tasks (annotated with the ID provided in Table 6.1), the rectangles with headings represent locations, the circles represent

Table 6.1: Work breakdown structure of the modelled activity

Task	ID
Load blocks into empty wheelbarrow	MW-01
Move blocks to wall construction site	MW-02
Unload blocks in wall construction site	MW-03
Load fine aggregate into empty wheelbarrow	MW-04
Move fine aggregate to mortar preparation site	MW-05
Unload fine aggregate in mortar preparation site	MW-06
Load cement bag into wheelbarrow	MW-07
Move cement bag to mortar preparation site	MW-08
Unload cement bag in mortar preparation site	MW-09
Load water into empty bucket	MW-10
Move water to mortar preparation site	MW-11
Prepare mortar	MW-12
Load mortar into empty bucket	MW-13
Move bucket with mortar to wall construction site	MW-14
Lay concrete block	MW-15

workers, the arrows represent the sequence of work, and the diamond represents the control structure, which is further expanded in Figure 6.3. As depicted in the figure, some tasks happen in a sequence (e.g. MW-01, MW-02 and MW-03), while others are performed without a specific, static order. In Figure 6.2, materials and equipment have not been modelled as separate entities for simplicity.

Figure 6.3 depicts the conceptual model of the selected activity again. This figure shows the expanded control structure of the system, which consists of two control units (CU-1 and CU-2). The control mechanisms of Control Units CU-1 and CU-2 are shown in Tables 6.2 and 6.3, respectively.

Table 6.2: Decision making mechanism of Control Unit CU-1 (from Figure 6.3)

Next node	Conditions
CU-2	`mortar volume in preparation site = 0`
MW-01	`blocks in wall construction site = 0` `blocks in storage site > 0`
MW-13	`bucket of mortar in wall construction site = 0` `mortar volume in preparation site > 0`
MW-15	`blocks in wall construction site > 0` `bucket of mortar in wall site = 1`

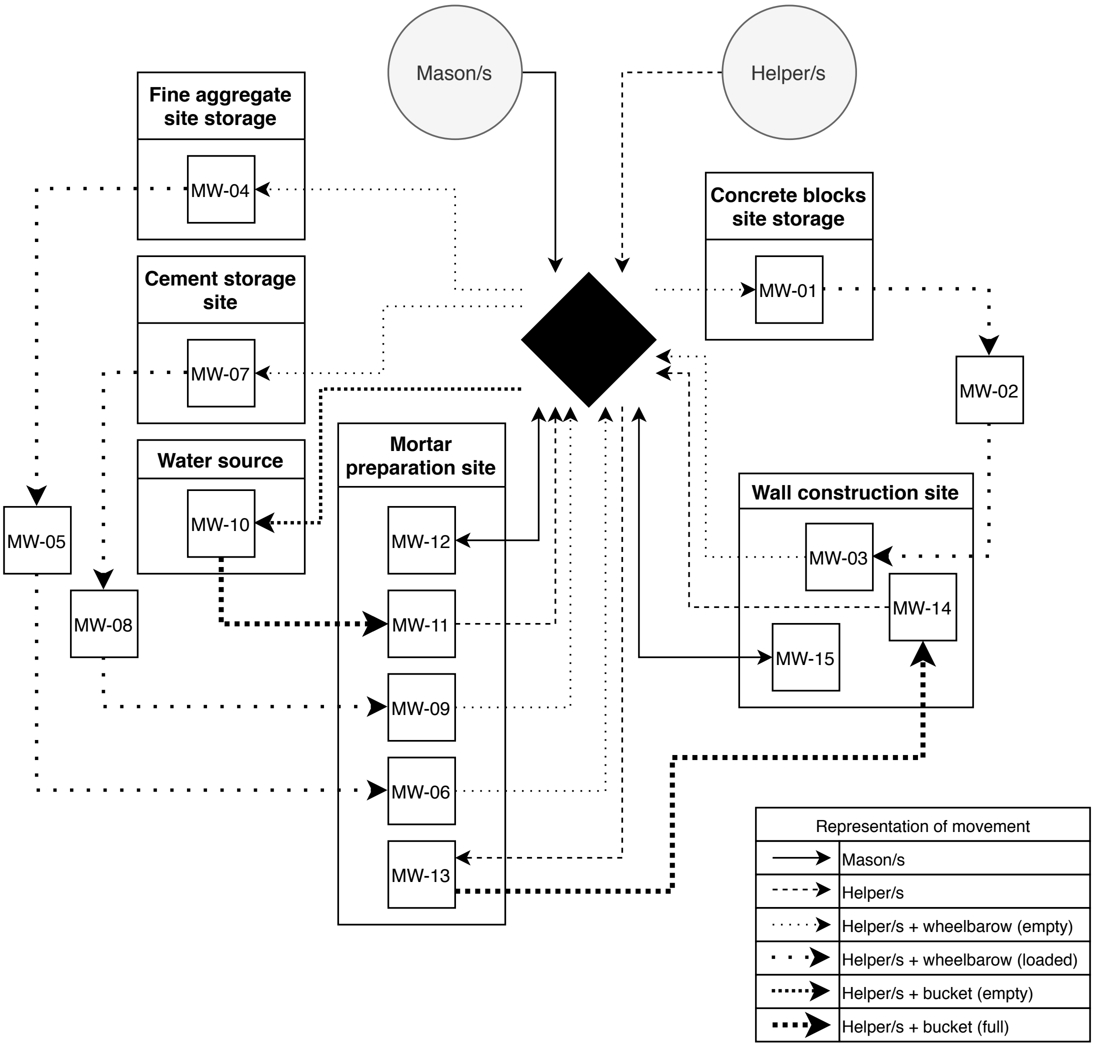


Figure 6.2: Conceptual model of the selected activity with locations

Table 6.3: Decision making mechanism of Control Unit CU-2 (from Figure 6.3)

Next node	Conditions
MW-04	`fine aggregate volume in mortar preparation site < 0.16`
MW-07	`cement bags in mortar preparation site < 1`
MW-10	`water buckets in mortar preparation site < 2`

MW-12	`fine aggregate volume in mortar preparation site` $\geq$ `0.16`
	`cement bags in mortar preparation site` $\geq$ `1`
	`water buckets in mortar preparation site` $\geq$ `2`

In Figure 6.3, the circles represent a step in the simulation in which the availability of the required workers and equipment is verified. If these resources are available, the task or block of tasks is executed. On the other hand, if the required resources are not available, the task or block of tasks is not executed nor scheduled. The

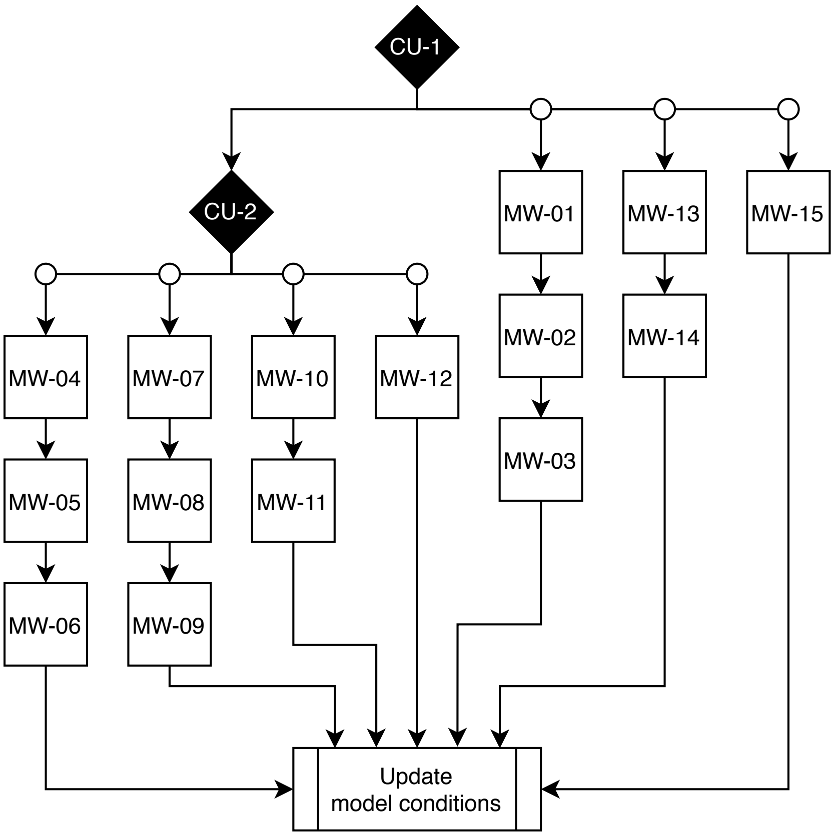


Figure 6.3: Conceptual model of the selected activity with control structure

reason for this is that changes in the conditions of the integrated model may render some of the scheduled tasks redundant. For example, if the task “prepare mortar” is scheduled, but another submodel executes a task in which mortar is prepared, there is no need to execute the scheduled task. With this modelling approach, tasks are only considered if all their requirements are met.

As Figure 6.3 shows, in this model entities only flow downstream from the control units. In this case, entities can be seen as signals that prompt the model to execute the tasks when all their conditions are met. After a task or block of tasks is executed, the conditions of the model are updated and new entities can flow from the distributed simulation component accordingly.

6.6.3 Duration of tasks

Table 6.4 shows the probability density functions of the durations of the simple tasks outlined in Table 6.1 alongside their parameters. These durations are given in minutes unless explicitly stated otherwise in the comments. Whenever a task is executed during simulation time, a random value of its duration is generated based on its functions and parameters.

6.7 Environment

This section details the development of the components of the environment module of the proposed framework following the implementation approach introduced in Section 5.2.

6.7.1 Constructable Element

As detailed in Chapter 5, Constructable Element interfaces were implemented as C# script game components. Figure 6.4 depicts the class diagram of the *Constructable Masonry Wall*, which extends the Constructable Element class.

The first five fields in Figure 6.4 make reference to the required materials to build the element based on its geometry. As discussed in Chapter 4, these properties determine the amount of materials that need to be instantiated to complete the model. The *GetRequiredMaterials* method is used to populate these properties.

The *firstBlock* field indicates the position of the first CMU of the wall within the 3D environment. The *courses* and *unitsPerCourse* fields are integers that determine, respectively, the number of courses or rows of blocks that comprise the wall, and how many blocks comprise each course. These properties are also

Table 6.4: Duration of masonry wall with concrete block tasks

Key	Tasks	Probability distribution	Parameters	Comments
D-01	MW-01 MW-03	Triangular	$a = 0.125$ $b = 0.625$ $c = 0.2$	Loading/unloading a concrete block
D-02	MW-02 MW-05 MW-08	Triangular	$a = 0.0125$ $b = 0.03$ $c = 0.02$	Moving loaded wheelbarrow (minutes per metre)
D-03	MW-04	Triangular	$a = 0.6$ $b = 1.2$ $c = 1.0$	Loading wheelbarrow using shovels
D-04	MW-06	Triangular	$a = 0.16$ $b = 0.5$ $c = 0.3$	Unloading wheelbarrow by overturning
D-05	MW-07 MW-09	Triangular	$a = 0.3$ $b = 0.9$ $c = 0.5$	Loading/unloading cement bag
D-06	MW-10 MW-13	Deterministic	$t = 0.5$	Filling bucket
D-07	MW-11	Deterministic	$t = 1.0$	Based on the assumption that there are multiple water sources on site
D-08	MW-12	Normal	$\mu = 12.562$ $\sigma = 5.1949$	Preparing mortar in mixer
D-09	MW-14	Normal	$\mu = 1.73463$ $\sigma = 0.35884$	Based on the assumption that the mortar is prepared no further than 100 m from the walls
D-10	MW-15	Weibull	$k = 2.01378$ $\lambda = 1.27307$	Laying a concrete block in place
D-11	Return	Triangular	$a = 0.01$ $b = 0.02$ $c = 0.0125$	Helper moving unloaded wheelbarrow (minutes per metre)

computed from the element's geometry, and their purpose is twofold. First, they are used to determine the position of the next block in the wall with the values of the *nextBlock*, *currentCourse* and *currentUnitInCourse* properties, which are updated during the simulation. Second, they are used to identify the CMUs at the edges of the wall that need to be resized based on the criteria shown in Figure 6.5.

Figure 6.6 shows a block at the edge of a wall within the 3D environment high-

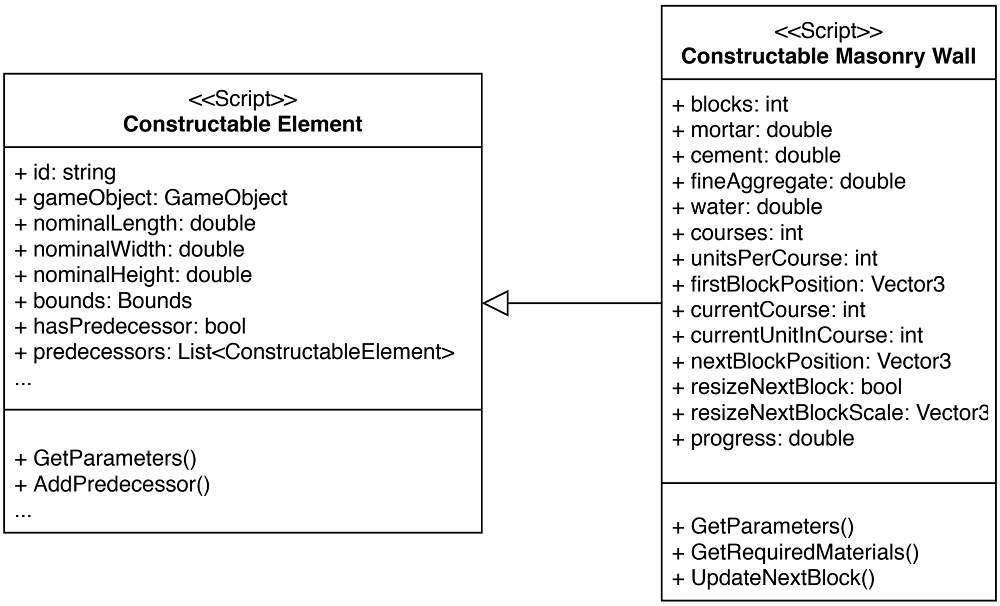


Figure 6.4: Constructable Element script for masonry walls

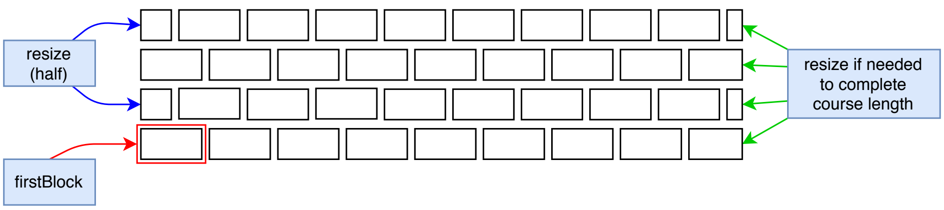


Figure 6.5: Criteria to resize CMUs at the edges of the wall

lighted in orange. Based on its position within the course where it was laid, this block would be resized. In the case presented in this section, block resizing has visualisation purposes only. However, it can be noted that by identifying individual pieces in this manner, the proposed framework can be used to estimate the number of blocks that require cutting, and the time and cost that this task would imply.

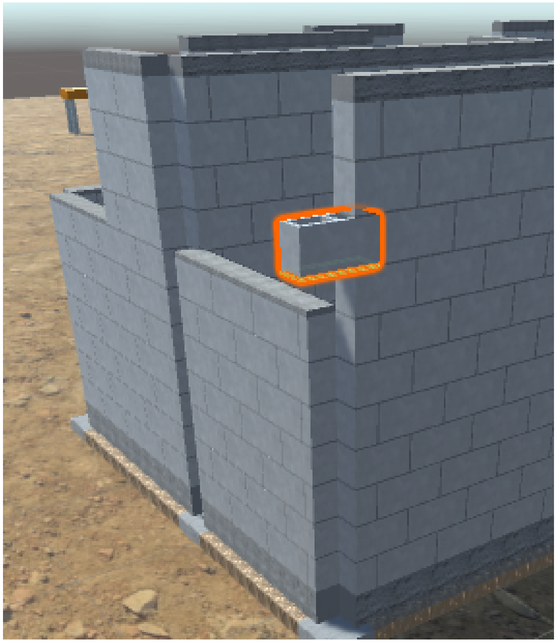


Figure 6.6: A block that needs to be resized

Construction progress of this activity, represented by the *progress* property, is measured by counting the number of blocks laid in place and comparing them to the total number of blocks required.

6.7.2 Generic simulation model of the construction activity

Based on the conceptual model outlined in Section 6.6, a generic simulation model for masonry walls with concrete blocks was developed following the implementation approach described in Section 5.2.2. This model consists of a C# script game component developed using the SharpSim library (Ceylan and Gunal 2011). The federate is composed of eighteen events. Fifteen events represent the activity's tasks documented in table 6.1. Two events represent the activity's control units depicted in Figure 6.3. The remaining event (labelled *R.R.* for resource released) updates the distributed simulation component when resources are released by tasks or blocks of tasks in the federate.

Figure 6.7 depicts the generic simulation model of this activity. In the figure, circles represent events, black arrows represent edges that link events within the submodel, and green arrows represent edges that link events in the submodel with an event in the distributed simulation component, labelled as *Main CU*. The latter type of edges are automatically generated by the system for each federate of the construction activity.

As detailed in Section 5.2.2, the simulation clock in SharpSim models is advanced by the interEventTime property of the *Edge* class. Furthermore, edges also act as valves that allow or prevent the flow of entities in the system based on predefined behaviour coded into the event's *State Change Listener*. Figure 6.8 depicts the flow diagram of the state change listener of the event MW-01 as an example of the edges' behaviour. In the figure, the interEventTime property of edges utilises the duration key provided in Table 6.4. As the figure shows, the duration of the edge between events MW-01 and MW-02 depends on the distance between the concrete block storage site and the corresponding wall construction site. Equally, before starting to load blocks, the helper needs to move an empty wheelbarrow to the block storage site. The duration of such movement depends on the distance between the wheelbarrow and the block storage site.

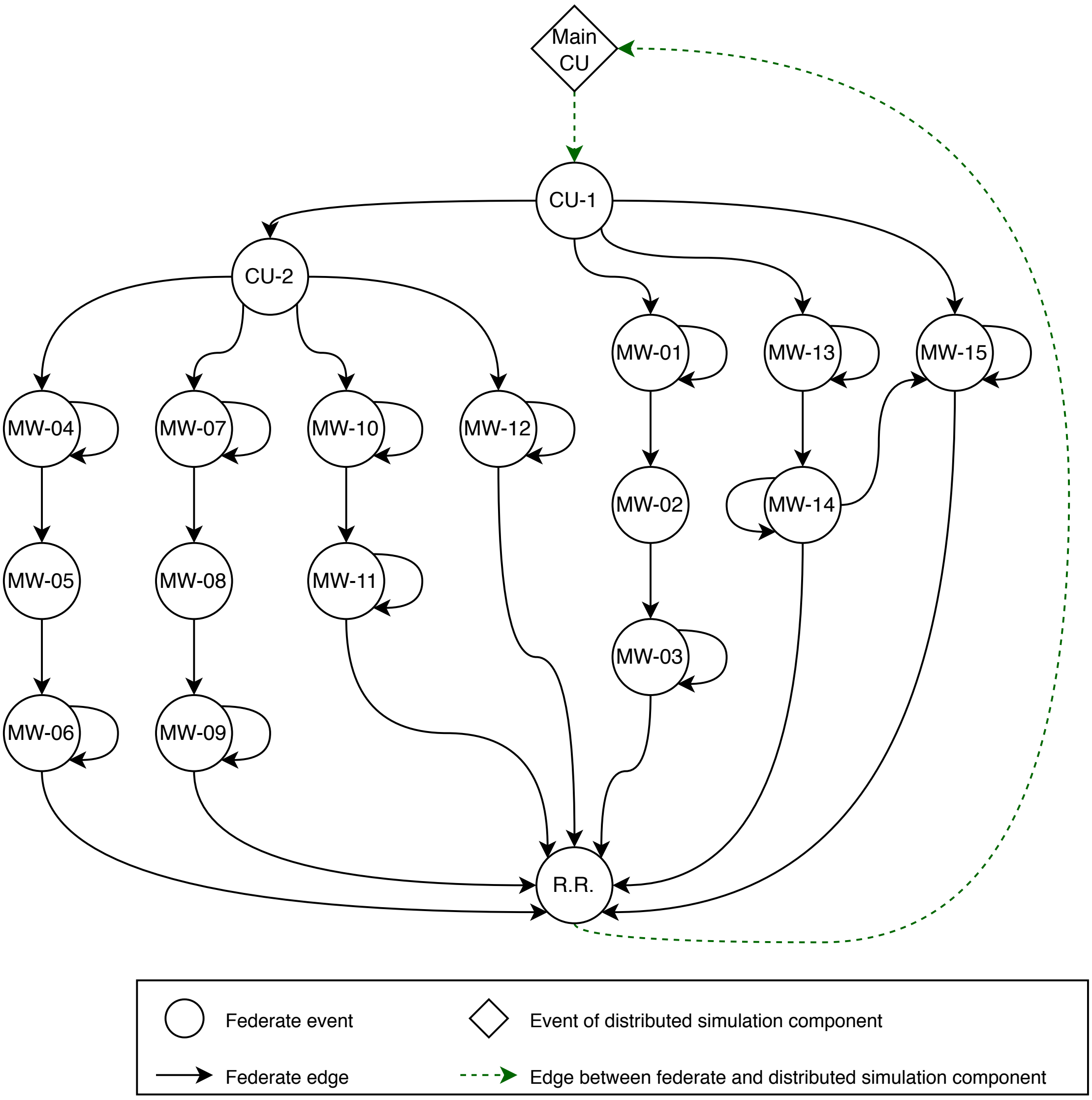


Figure 6.7: Generic simulation model of the selected construction activity

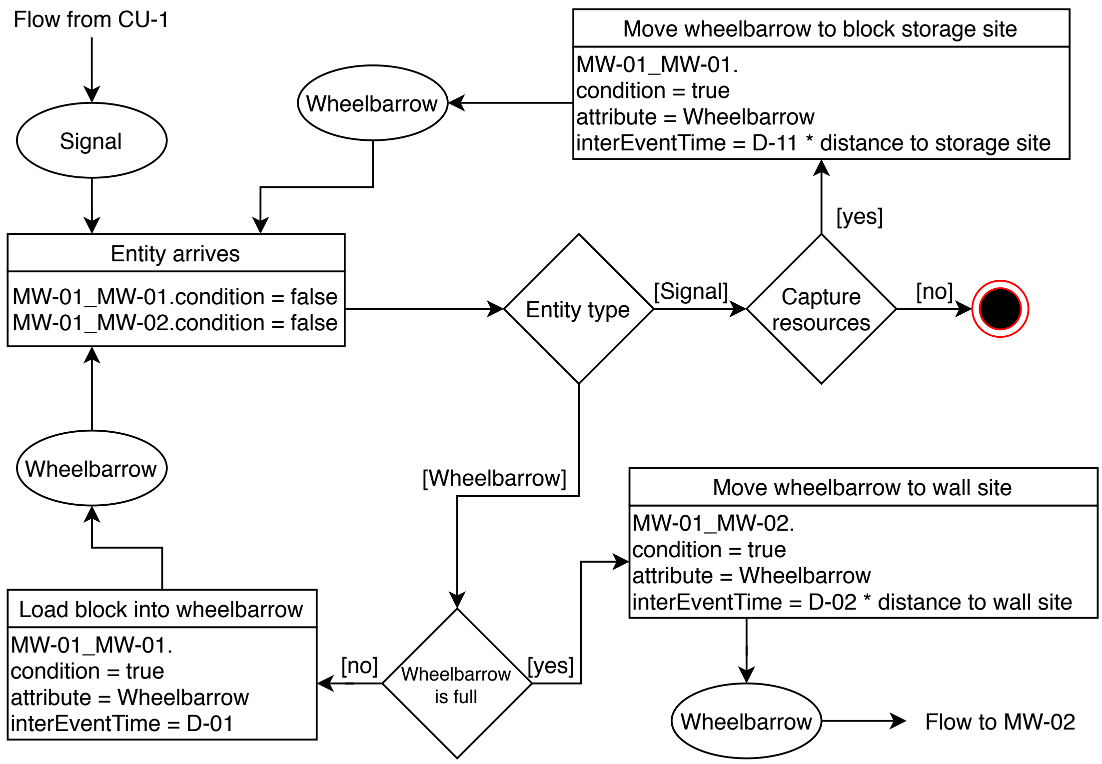


Figure 6.8: Flow diagram of the state change listener of event MW-01

6.7.3 Models of resources

As discussed in Section 5.2.3, resources of the material type are instantiated at runtime during simulation. In this model, all the required materials are instantiated at the beginning of the simulation. Game objects representing such materials are located at the storage site position as indicated in Figure 6.1. These game objects are instantiated from predefined reusable assets loaded into the scene. Figure 6.9 shows the storage site with the required materials to complete the masonry walls of one house.

Equally, resources of the equipment and worker types are instantiated from preloaded reusable assents. However, the number of available resources of these types available during the simulation is user defined. The objective of the case study was to determine the number of helpers that a mason needs to keep a continuous

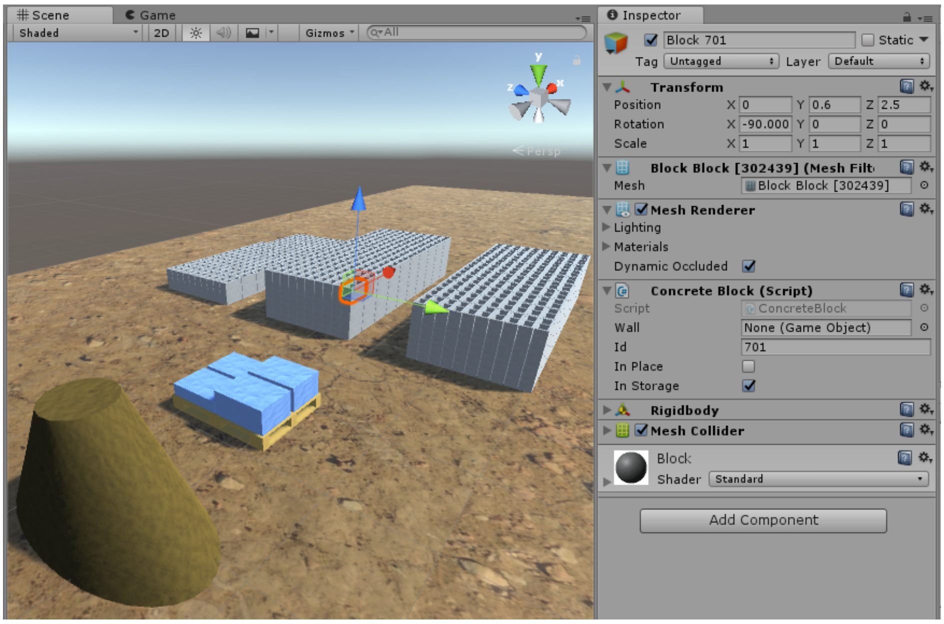


Figure 6.9: Materials instantiated in the storage site

workflow based on the distance between the house and the storage site. Thus, the construction activity was simulated for a randomly selected house from each zone using crews of a mason and a different number of available helpers. The number of available wheelbarrows matched that of the helpers for each scenario. A single concrete mixer was also loaded into the scene.

6.8 BIM model

A BIM model of the house prototype that was built in the selected project was imported into the scene as an fbx file, as detailed in Section 5.3.1. Once in the scene, a script of the Constructable Element type and a script of the generic simulation model type were attached to each game object representing masonry walls. The model is composed of 21 masonry walls. Figure 6.10 shows the BIM model loaded into the scene. Both attached scripts can be seen on the inspector window on the right side of the figure for the highlighted wall.

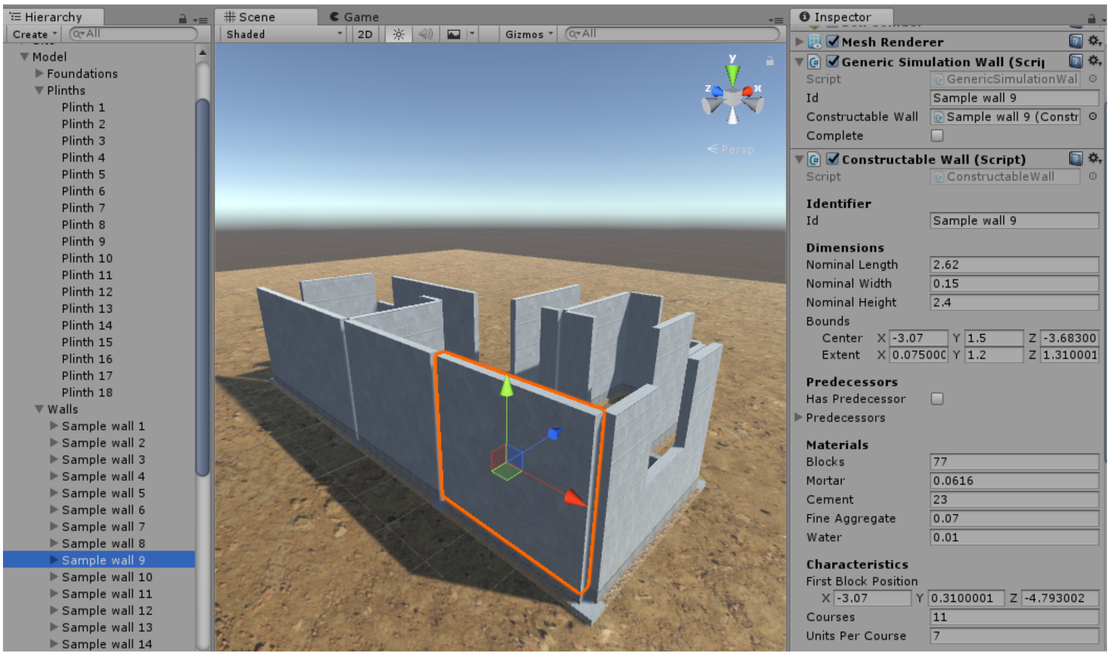


Figure 6.10: BIM model of masonry walls loaded into scene

6.9 Simulation model

The simulation model of the selected construction activity for the input BIM model consists of 21 federates based on the generic simulation model depicted in Figure 6.7. As the figure shows, each federate is linked to the *Main CU* event of the distributed simulation component. This event represents the control unit at the topmost level of the control structure hierarchy.

Listing 6.1 shows the State Change Listener of the *Main CU* event. This is the first event that is executed when the simulation starts. Subsequently, it sends a signal to each incomplete wall federate. As shown in Figure 6.7, this event is executed again every time that a resource becomes available.

```
void MainCU(SharpSim.Event evt)
{
    evt.EventExecuted += delegate (object obj1, EventInfoArgs e)
    {
        // All edges from the Main CU event are disabled
        foreach(Edge edge in edgesList)
        {
            if (edge.targetEvent != mainCU)
            {
                edge.condition = false;
            }
        }

        // Edges from the Main CU event to incomplete wall submodels
        // are activated
        foreach(GenericSimulationWall wall in wallSubmodels)
        {
            if(!wall.complete)
            {
                edgesList.Find(
                    edge => edge.name == "Link to wall " + wall.id).
                        condition = true;
            }
        }
    };
}
```

Listing 6.1: State Change Listener of the main control unit of the distributed simulation component

6.9.1 Modelled scenarios

As previously stated, the model seeks to determine the number of helpers that a mason needs to keep a continuous workflow based on the location of the house within the zones depicted in Figure 6.1. Table 6.5 shows the different scenarios that were simulated to achieve the objective of the case study. Each scenario was simulated twenty times and the results were compared to provide recommendations. A Kolmogorov-Smirnov goodness-of-fit test was applied to the results to ensure that the number of simulation runs was enough to produce a normal distribution. Three simulation outcomes were observed in order to conclude about the different crew compositions: 1) the average duration of the construction activity, 2) the average waiting time of the mason, and 3) the average walking distance per day of helpers.

Table 6.5: Modelled scenarios

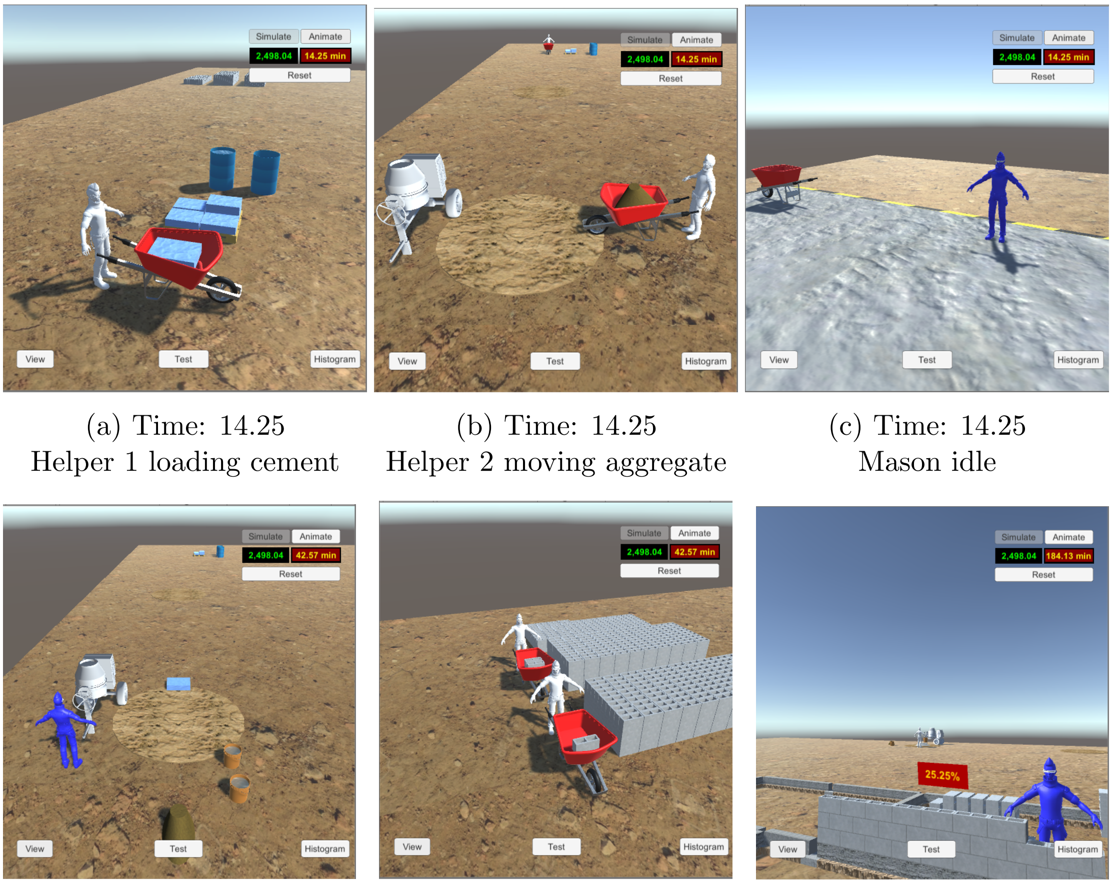
Scenario	**Zone**	**Average distance to storage site *(m)***	**Number of masons**	**Number of helpers**
1	4	670.93	1	1
2	4	670.93	1	2
3	4	670.93	1	3
4	4	670.93	1	4
5	4	670.92	1	5
6	3	470.93	1	1
7	3	470.93	1	2
8	3	470.93	1	3
9	3	470.93	1	4
10	2	320.93	1	1
11	2	320.93	1	2
12	2	320.93	1	3
13	1	120.93	1	1
14	1	120.93	1	2

6.9.2 Model validation

The model was validated using a combination of two techniques, namely, animation and traces. The animation technique consists of graphically visualising the operational behaviour of the model as it moves through time, while in the traces technique the behaviour of different types of entities in the model are followed through the model to determine it its logic is correct (Sargent 2013). Validation was performed by visualising the simulation-based animation of a randomly selected run of each scenario. Figure 6.11 shows the location of workers from a random run of Scenario 2 at different times.

6.9.3 Simulation results

This case study did not attempt to draw generic conclusions from the obtained results that favour a crew composition over another. Its purpose was to demonstrate



(a) Time: 14.25
Helper 1 loading cement

(b) Time: 14.25
Helper 2 moving aggregate

(c) Time: 14.25
Mason idle

(d) Time: 42.57
Mason ready to prepare mortar

(e) Time: 42.57
Both helpers loading concrete blocks

(f) Time: 184.13
Mason waiting for mortar
Helper 1 loading mortar

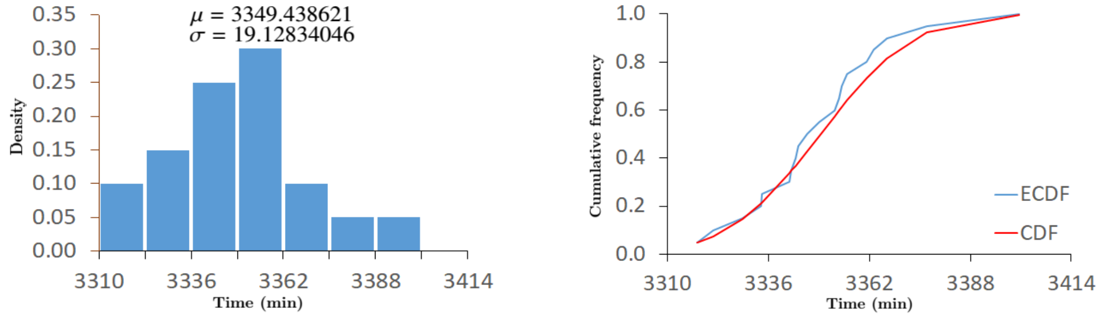
Figure 6.11: Resources at different points in time in Scenario 2

that the implemented framework enables the comparison of the performance of a construction project with different conditions without re-formalising the simulation model. This section presents the results of the observed simulation outcomes to illustrate the employment of the proposed framework.

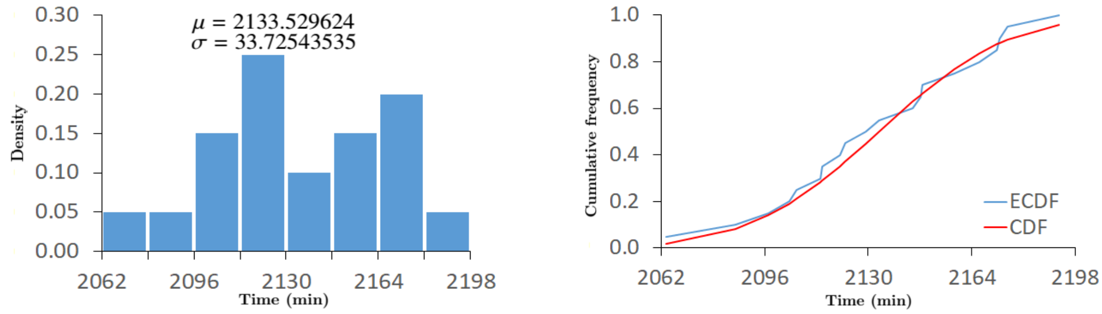
Figure 6.12 illustrates the resulting durations obtained from the scenarios of Zone 3. The graphs on the left side of the figure show the frequency with which the durations were observed throughout the simulation runs, while those on the right side show a comparison between the empirical cumulative distribution function (ECDF) of the observed results and the normal cumulative distribution function (CDF). As shown by the figures, the obtained results are normally distributed. Additionally, a Kolmogorov-Smirnov goodness-of-fit test was applied to the duration data sets. Results show that, as expected, all the duration data sets follow normal distributions with a significance level of $\alpha = 0.05$.

Table 6.6 reports the means of the three simulation outputs observed for each modelled scenario analysed in this case study. These results provide valuable information to support informed decision-making. For example, in Zone 4 the estimated completion date of the activity performed by a crew of a mason and a single helper (Scenario 1) occurs on the ninth day, considering working days of eight hours. In contrast, in the same zone, both estimated completion dates of the activity performed by crews of a mason and four helpers (Scenario 4) or a mason and five helpers (Scenario 5) occur on the fourth day. Equally, the average daily walking distance of helpers in Scenarios 4 and 5 is reduced by 43.33% and 54.34%, respectively, compared to that of Scenario 1.

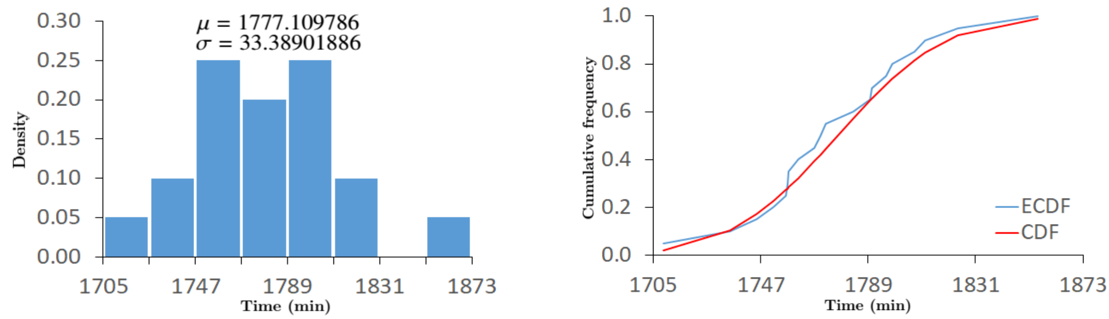
Figure 6.13 depicts a comparison between two equal time segments of the resource histograms of random runs of Scenarios 2 and 4. Coupled with the animations, these reports provide sufficient insights to support an informed decision-making.



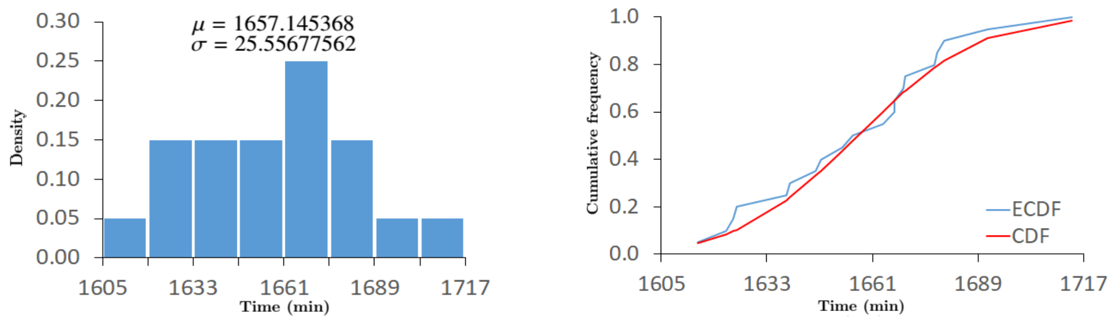
(a) Scenario 6



(b) Scenario 7



(c) Scenario 8



(d) Scenario 9

Figure 6.12: Simulation results from the scenarios of Zone 3

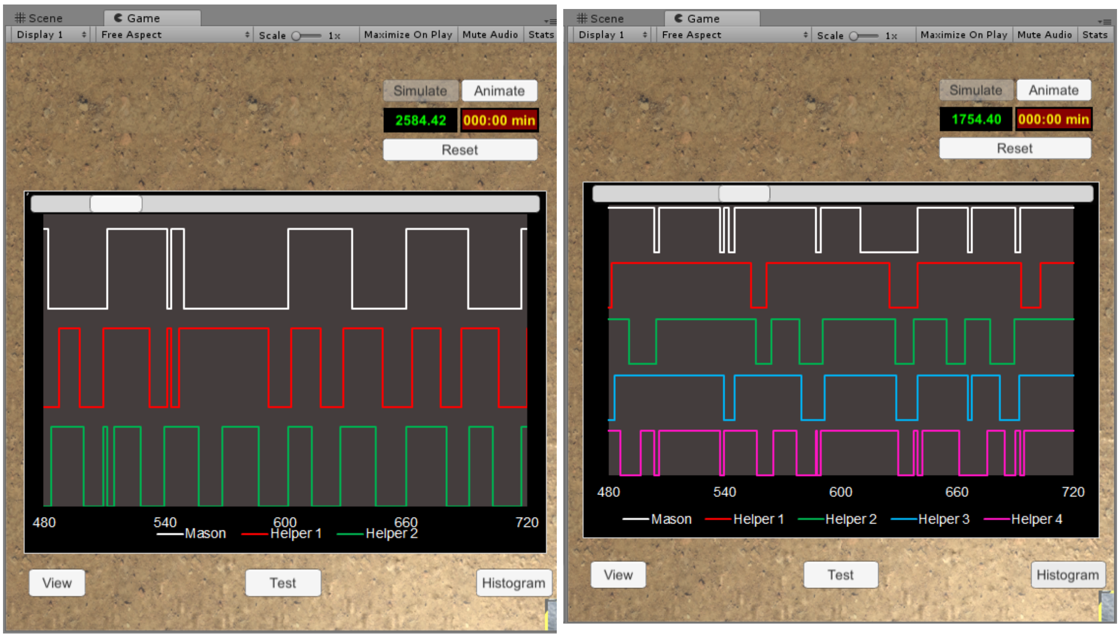
Table 6.6: Simulation results

Zone	Scenario	Duration *(min)*	Waiting time of mason *(min)*	Walking distance per helper *(Km/day)*[a]
4	1	4256.61	1644.22	12.44
	2	2519.07	404.74	9.53
	3	1947.24	112.40	8.00
	4	1804.36	47.87	7.05
	5	1711.72	45.45	5.68
3	6	3349.44	1149.85	10.85
	7	2133.53	208.81	7.92
	8	1777.11	44.67	6.60
	9	1657.15	25.71	4.96
2	10	2581.16	622.17	8.87
	11	1778.35	89.50	6.75
	12	1611.93	22.72	4.47
1	13	1802.66	201.62	4.90
	14	1522.13	24.67	2.49

[a] Considering working days of eight hours.

For example, Table 6.6 also shows that in both Scenario 4 and 5, the average waiting time of the mason was similar, although the mean estimated duration is approximately 1.5 hours shorter in the latter. When comparing the simulation-based animations of these scenarios, it is observed that this difference is due to the fact that when the mason requires preparing mortar, the materials arrive faster to the mortar preparation site in Scenario 5, thanks to the availability of more helpers. With such insights, the model can support decision-makers to decide whether the cost of larger crews is worth gaining approximately 1.5 hours per house in Zone 4.

Figure 6.14 shows a comparison of the performance of the construction activity in the different modelled scenarios. In the figure, the performances of the construction activity in the modelled scenarios of a given zone are compared to that of the minimum crew scenario for that same zone.



(a) Scenario 2 (b) Scenario 4

Figure 6.13: Comparison between histograms of two scenarios

6.10 Remarks

The results of the discussed case study show how small changes in the composition of the crews lead to a significant impact on the performance of a construction activity. This case study demonstrates the usefulness of simulation studies to support planning. The proposed framework enabled the automatic generation of a simulation model based on an existing BIM model. This simulation model considered project-specific constraints (i.e. the distance between the construction site and the material storage site) and allowed a quick scenario development for what-if analysis. The simulation-based animations were used firstly to validate the model and, secondly, to compare resource usage between two different scenarios. This process provided a better understanding of the modelled case study.

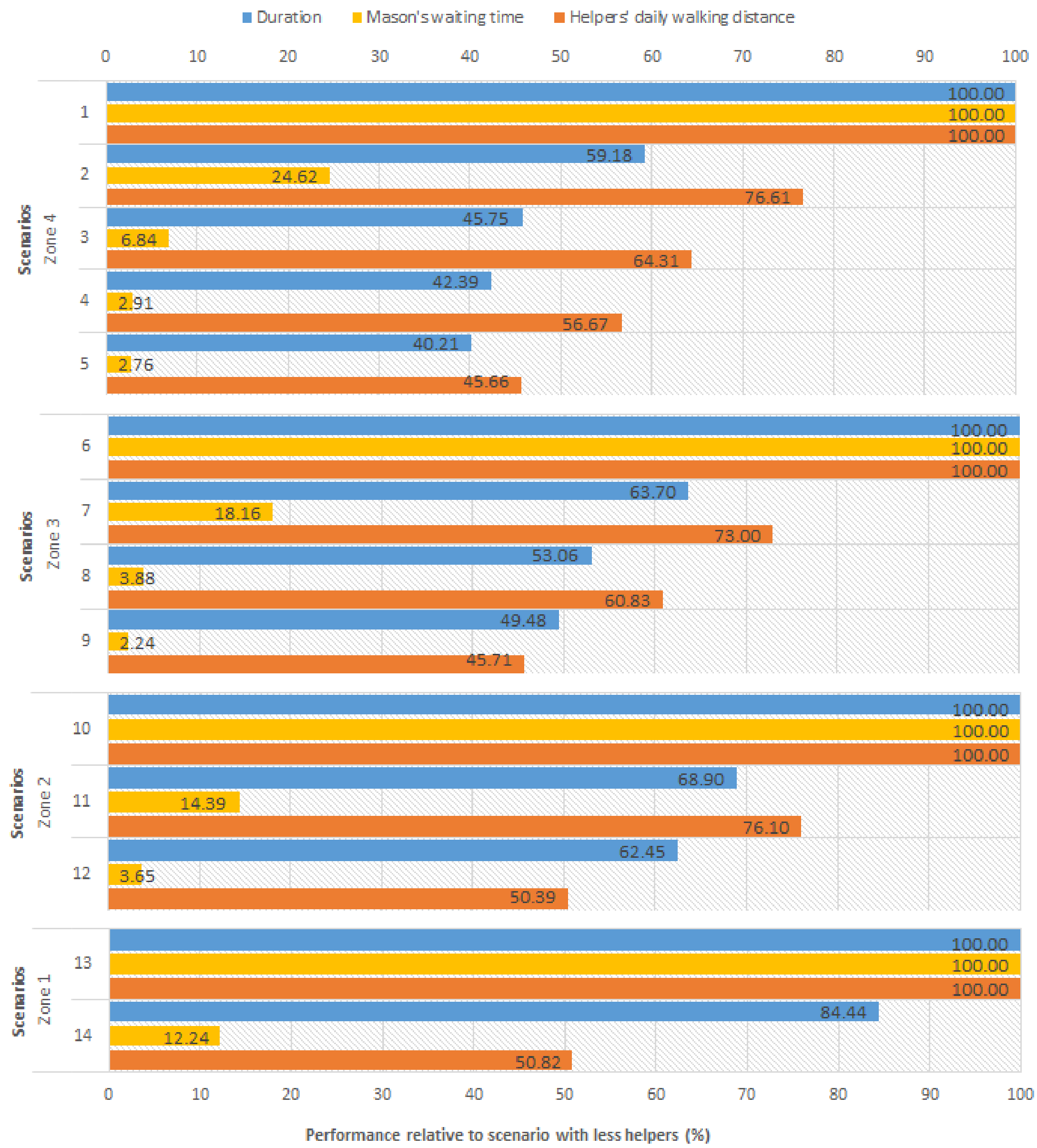


Figure 6.14: Comparison of the performance of the construction activity in the different modelled scenarios

6.11 Summary

This chapter assessed the employment of the implemented framework for the integration of construction simulation and BIM through a case study based on a typical masonry construction problem. The system recognised BIM elements representing masonry walls and assigned to each one an appropriate Constructable

Element script. The Constructable Element scripts effectively extracted parameters from their corresponding BIM elements. Subsequently, the system instantiated appropriate simulation models based on the generic simulation model of the selected activity and the information contained on each Constructable Element script in the scene. Different scenarios based on the resulting simulation model were developed to support the decision-making process of the selected project in terms of masonry crew composition. Most notably, the simulation models were not re-formalised at any point during scenario development. Simulation-based animations were produced for each simulation run, allowing for an enhancement of the visualisation of results. The results of the case study reveal that the selected game engine allowed to achieve the sought functionality of the proposed framework. Furthermore, the case study showcased the usefulness of the framework as a planning and decision-making support tool.

This chapter also illustrated how the Constructable Element and the generic simulation model for the selected construction activity, as well as scenarios for what-if analysis, were developed. Thus, this chapter provides guidance to users to develop customised environments based on the proposed framework. The next chapter presents a summary of the research and concludes the thesis.

Chapter 7

Conclusions and recommendations

This chapter summarises and concludes the thesis. It discusses how the objectives of the research were achieved and provides conclusions relevant to those objectives. The chapter also recapitulates the main contributions of this work to the current body of knowledge and lists publications produced in connection with this research. It also discusses the limitations of the work. In addition, the chapter provides recommendations for future work to further extend the research.

7.1 Research summary

Although construction simulation has been proven to be effective as a planning and decision-making support tool, it has not been fully embraced by industry practitioners. Therefore, the recognised benefits of construction simulation have not been exploited. On the other hand, the academic community in the field of construction engineering and management has valued simulation as an essential tool to understand the behaviour of construction systems and to carry out comparative research studies.

Recent research efforts have attempted to identify the reasons for the gap between research and current industry practice in regards to construction simulation. The difficulty of developing complex simulation models with respect to the amount of data, skills, effort and time that this process entails has been identified as one of the challenges hindering industry adoption of simulation. Equally, the fact that these hardily-developed models are usually single-use and cannot be capitalised by reusing them across multiple projects also discourages decision-makers from devoting resources to model development. Moreover, simulation results are often presented in such a way that only those familiar with this method can easily interpret them, forcing decision-makers to separate themselves from the process. In turn, simulation results are either ignored or misapplied, resulting in a lack of confidence in the method.

BIM models have been suggested as a means to streamline the simulation model development process while enabling model reuse to a certain extent. On the other hand, simulation-based animations are regarded as practical tools to communicate complex models and their corresponding results to domain experts that are unfamiliar with simulation. However, current simulation approaches have failed to leverage BIM to develop simulation models that consider resource allocation and task interdependencies constraints while providing a mechanism to produce animations that enhance the visualisation of the simulation model and its results. Therefore, this research investigated how BIM models can be integrated into the simulation model development process without overlooking fundamental constraints found in construction projects. The proposed approach aimed to enable model reuse and to provide a means to enhance results visualisation.

The study investigated modelling and simulation approaches, the challenges that prevent their adoption in the construction industry, and the aspects that could assist in overcoming those challenges. Based on this review, the research proposes an integrated simulation approach, described in Chapter 4. The proposed strategy consisted of a distributed simulation in which construction activities were modelled using a hierarchical control structure. This approach provides a mechanism to deal with interdependencies between tasks of different activities and allocating shared resources and enables model reuse. The construction activity federations in the distributed simulation were parametric simulation models, which reduce the extensive manual modifications that simulation models are typically subject to when small changes occur in the project. As such, the models are able to represent a variety of products of the same type based on product parameters and project-specific constraints inherent to construction, thus, further enabling model reuse. For that reason, construction activity federations were called *generic* simulation models.

The role of BIM in the proposed simulation approach was to provide input parameters to these generic simulation models of construction activities. To this end, an interface between the BIM element and its corresponding generic simulation model, called *Constructable Element*, was developed. The purpose of the Constructable Element interface was to extract and enrich data from BIM elements and translate it into the requirements of generic simulation models.

The proposed simulation approach also adopted the post-processing visualisation technique, which consists of generating time-stamped *trace files* at each update of the modelled system during each simulation run. These race files indicate the position and state of materials, workers and equipment during each step of the construction process of the simulated project. Therefore, a collection of chronologically ordered set of trace files represents a detailed log of what happened during its corresponding simulation run. The main purpose of these trace files was to provide a means to produce an animation that depicts accurately how the model changed in time during the simulation run, getting into the details of resource interaction. Such animations can be used for several visualisation purposes, including model validation and verification.

Based on this integrated simulation approach, the research proposed a framework for the integration of BIM and construction simulation, detailed in Chapter 4. The components of the framework were designed to respond to the proposed integrated simulation approach, thus, facilitating simulation model development using BIM, enabling model reuse and enhancing results visualisation. The framework consists of five main modules. (1) The environment module is preloaded with a library of Constructable Elements for different products, a library of generic simulation models that represent the construction activities to build such products, and a library of 3D models of resources for visualisation in the simulation-based animations. (2) The user input module provides the facility to import existing BIM models into the system. (3) The pre-processing module automatically develops a simulation model of the construction project based on its BIM model. This is accomplished by assigning a Constructable Element and a generic simulation model to each BIM component of interest based on its type. (4) The simulation module provides an environment to experiment with different construction alternatives and resource constraints without the need to re-formalise the simulation model. Furthermore, it allows integration of other independently-developed simulation models due to the distributed simulation approach. Such models could, for example, represent weather, labour absenteeism, equipment breakage and maintenance, material procurement processes, etc. (5) The visualisation module produces different reports for planning and decision-making support as well as animations for diverse visualisation purposes. The reports include resource histograms, project duration, productivity at various levels and schedules.

The proposed framework for the integration of BIM and construction simulation was implemented within a game engine. Chapter 5 describes the details of the development of a tool based on such an implementation. Several game engine related concepts were leveraged to achieve the sought functionality. *Game objects* are the fundamental objects that represent characters, props and scenery. *Game components* implement the functionality of game objects. *Scripts* are a type of game component that consists of a piece of code that implements the developer's own features. *Reusable assets* are templates of game objects and their game components from which new instances of the game object can be created.

The application described in this research was developed in the Unity game engine. The Constructable Elements and the generic simulation models were implemented a C# script game components. For the latter, an existing C# open-source discrete-event simulation code library, SharpSim, was used. Equally, the Math.Net Numerics library was used to model the durations of tasks within the SharpSim models.

BIM models can be imported into the scene as an fbx file or as an IFC data model. Once imported, each BIM element of the model becomes an individual game object. Game components of the Constructable Element and generic simulation model classes are instantiated and attached to BIM element game objects in the scene based on their type. Once this process is complete, the simulation models in the scene are no longer generic, since they represent the construction activities required to build the BIM elements associated with them. Subsequently, the distributed simulation component, also implemented as a C# script game component, can include the construction activity federation to the federate.

Trace files were also implemented as C# script game components following an object oriented approach. A new object of the trace file class, which contains all the relevant information to produce reports and animations, is instantiated upon updates in the modelled system.

This thesis assessed the feasibility of the implemented framework through a case study based on a typical masonry construction problem, described in Chapter 6. The case study analysed several compositions of crews performing the construction of masonry walls with concrete blocks in a large housing project. The project was divided into four zones based on the distance between the houses and the materials storage site. The selected construction activity was simulated for a randomly chosen house from each zone using crews of a mason and a different number of available helpers. The model was validated using a combination of animation and traces by visualising the simulation-based animation of a randomly selected run of each scenario and following the behaviour of workers.

The presented case study underlined the usefulness of simulation studies to support planning and decision-making. Furthermore, it demonstrated that the proposed framework enables the automatic generation of a BIM-based simulation model, and reduces the skills, data, time and effort required to develop a traditional simulation model. The resulting simulation model considered project-specific constraints, such as the distance between the construction site and the material storage site, and allowed a rapid scenario development for what-if analysis. The simulation-based animations were used firstly to validate the model and, secondly, to compare resource usage between two different scenarios. The animations provided a better understanding of the modelled case and the simulation results.

Finally, this study also illustrated the development of the required components of the framework and provided a guide on how to design customised environments based on the proposed framework.

7.2 Achievement of aim and objectives

The primary aim of this research was to investigate how to use BIM models to streamline construction simulation model development without overlooking constraints inherent to construction projects, particularly, resource allocation and task interdependencies constraints. The development of the framework for the integration of BIM and construction simulation, as well as the proposed integrated simulation approach that underpins such a framework, satisfy this primary aim.

This thesis shows that:

1. The proposed framework semi automatically develops a simulation model based on an existing BIM model.
2. The framework provides a suitable environment to experiment with different construction alternatives and resource constraints without the need to reformalise the simulation model.
3. The framework is capable of producing simulation-based animations that enhance the visualisation results and that can be used for diverse visualisation purposes.
4. The proposed integrated simulation approach can be leveraged for model reuse.
5. A game engine is a suitable environment to implement the proposed conceptual framework.
6. The implemented framework is reliable to solve a typical construction management problem.

The primary aim of this research was also achieved through the completion of the research objectives outlined in Section 1.3. This section summarises how such objectives were achieved.

Objective 1: To review existing simulation approaches used in the construction context and existing techniques applied to visualise simulation results

This work contributed to respond to the following research questions: *Q1) How is simulation used in the construction context?*, *Q2) What are the current trends in the field of construction simulation?*, and *Q3) How are constraints related to resource allocation and task interdependencies represented in construction simulation models?* Research into construction simulation, its uses, tools and limitations, presented in Chapter 2 revealed the current practice and trends in the field of simulation in the construction context. More specifically, the current trend on facilitating the simulation model development process using BIM was identified as a promising approach to overcome the challenge posed by the amount of data, skills, effort and time that this process entails. Hierarchical control structures were identified as a potential solution to represent constraints inherent to construction projects. These control structures coupled with the distributed simulation approach adopted in this research are able to represent allocation of shared resources and interdependencies between tasks of different construction activities. During the development of the simulation approach presented in Chapter 4, simulation model reuse was also addressed from the conceptual phase of simulation modelling by adopting the parametric modelling paradigm. Finally, animations were identified as an effective means to enhance the visualisation of simulation results, which is another identified challenge of simulation. The work presented in Chapters 4 - 5 details how the proposed framework implements post-processing visualisation to produce simulation-based animations. More specifically, Chapter 5 details how a game engine was leveraged to produce such an animation.

Objective 2: To design a framework to generate a simulation model from a BIM model and show its results in the form of animations

This work contributed to respond to the following research questions: *Q4) How can model reuse be leveraged in the simulation domain?*, and *Q5) How can the visualisation of simulation results be enhanced?* Chapter 4 describes the work that satisfied this objective. Supplemented by the work presented in Chapter 2, a framework for the integration of construction simulation and BIM was proposed. Such a framework enables the semi automatic development of a BIM-based simulation model by using BIM data as input parameters for generic simulation models that represent the construction activities required to build an associated BIM element. It also enables model reuse due to its simulation approach. The framework has a mechanism to present simulation results in an enhanced way through simulation-based animations.

Objective 3: To develop a game engine-based application based on the proposed conceptual framework that enables achieving its sought functionality

This work contributed to respond to the following research question: *Q6) How can the features of a game engine be leveraged to achieve Chapter 5 details how this objective was satisfied by developing an application based on the proposed frame-*

work within the Unity game engine. C# script game components were developed to achieve the sought functionality of the proposed framework. the sought functionality of the proposed framework? The features of the Unity game engine were investigated, and the way in which these features were integrated into the proposed conceptual framework was described.

Objective 4: To assess the usefulness of the implemented framework through a case study

This work contributed to respond to the following research question: *Q7) How reliable is a construction simulation model developed based on existing BIM data in solving a typical construction management problem?* This objective was achieved by developing a case study on a typical masonry construction problem. The case study, presented in Chapter 6, demonstrated that the framework implemented in a game engine meets its theoretical functionality. A simulation model based on an existing BIM model was automatically developed by the system. The application provided a suitable environment to rapidly develop multiple scenarios to analyse the case study. The system produced simulation-based animations that were used to verify and validate the automatically-developed model and to present simulation results. It is worth noting that the implementation of the system has some limitations, discussed in more detail in Section 7.6. Primarily, developing a comprehensive library of generic simulation models of construction activities and of Constructable Elements in the environment module of the framework is time consuming and, therefore, represents a difficult challenge. However, as the libraries are built up, developing more complex simulation models would become easier, as more activities would be available to streamline the process.

7.3 Revisiting the hypothesis

The hypothesis tested by this research was as follows:

A BIM-based construction simulation modelling approach can allow a semi-automatic development of complex construction simulation models by enabling model reuse and providing a means to visualise simulation results.

The research hypothesis was decomposed in seven research questions, mapped to the research objectives described in the previous sections.

It can be concluded from the above remarks that the hypothesis is true. The proposed approach allows a semi-automatic development of construction simulation models. However, the implementation of the system requires investment in the development of large libraries of generic simulation models of construction activities and of Constructable Elements in the environment module of the framework. While this limits the immediate practical application of the proposed conceptual framework, the process would become easier as these libraries are expanded and built upon.

7.4 Contribution to knowledge

The novelty of this research lies mainly in the development of the proposed framework for the integration of construction simulation and BIM. Equally, this research demonstrates that a game engine is a suitable environment to implement the proposed framework. A game engine-based artifact developed based on the proposed conceptual framework was applied to solve a typical construction management problem to assess its functionality.

The integrated simulation modelling approach that underpins the development of the proposed framework leverages a distributed simulation approach coupled with hierarchical control structures and parametric simulation models, which take input from existing BIM data.

The proposed framework enables model reuse and considers resource allocation and task interdependencies constraints. Furthermore, it is capable of semi automatically generating a construction simulation model based on a user input BIM model, which can be used to experiment different construction strategies and resource allocations for planning and decision-making support, as well as for research applications.

The implemented framework provides a mechanism to validate and verify the semi-automatically-developed simulation model and a means to present simulation results for visualisation purposes. It consists of producing simulation-based animations that follow a post-processing approach.

The research also demonstrates that the proposed framework is useful as a planning support tool by presenting an illustrative case study, which also serves as a guide to users on how to implement the framework and develop customised environments based on the work presented in this document.

7.5 Research dissemination

The following publications were produced in connection with the work presented in this thesis:

- Carlos Arturo Osorio-Sandoval et al. (2017). 'Computer technology for serious games in education: A literature review'. In: *Proceedings of the 24th EG-ICE International Workshop on Intelligent Computing in Engineering*

2017. EG-ICE, pp. 239–249

- Carlos Arturo Osorio-Sandoval, Walid Tizani, Christian Koch and Abdelaziz Fadoul (2018). 'Integration of building information modelling and discrete event simulation within a game engine'. In: *17th International Conference on Computing in Civil and Building Engineering*

- Carlos Arturo Osorio-Sandoval, Walid Tizani and Christian Koch (2018). 'A method for discrete event simulation and building information modelling integration using a game engine'. In: *Advances in Computational Design* 3.4, pp. 405–418

- Carlos Arturo Osorio-Sandoval et al. (2020). 'Discrete-event simulation and building information modelling based animation of construction activities'. In: *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering. ICCCBE 2020*

- Carlos Arturo Osorio-Sandoval et al. (2021 (tentative): Manuscript in review). *A framework for the integration of building information modelling and construction simulation*

7.6 Limitations

This section discusses some of the limitations of the work presented in this thesis.

- Extending the libraries of Constructable Elements and generic simulation models of construction activities in the environment module of the framework to implement it in different types of projects represents a difficult challenge. This situation makes impractical the wider application of the proposed framework without significant previous work. However, as the libraries are built up, developing more complex simulation models would become easier, as more activities would be available to streamline the process.

- The proposed framework lacks a mechanism to evaluate and optimise the locations of material storage sites and, therefore, relies entirely on user input to determine them. Nevertheless, the implemented system can aid decision-makers to find suitable locations of material storage sites by comparing the performance of the project with different scenarios in terms of such a variable. While this implies that users would need to devote time to develop such scenarios, the illustrative case study presented in Chapter 6 demonstrates that this process is not time-consuming and that the insights that the system provides are helpful to support decision-making.

- Equally, the framework is not able to evaluate the paths of resources moving within the construction site. This limitation hinders the possibility to use the system to improve safety by flagging plausible paths that may result in collisions. This is particularly important when the paths of heavy machinery and human workers intersect because being struck by a moving vehicle or heavy equipment is one of the most frequent causes of work-related casualties in the construction industry.

- The simulation-based animations produced in the visualisation module of the proposed framework only show the location of resources at the discrete points in time in which events occurred and triggered the instantiation of a trace file. Consequently, the continuous motion of resources is not available. However, the source and target points, as well as the start and finish times of the motion are available in the trace files. Therefore, there is potential to overcome this limitation by using these data to model the missing continuous motion. Although resource paths would not be optimised, this solution may also serve to detect the hazardous paths of resources discussed above.

7.7 Recommendations for future research

This section outlines recommendations for future research, including, but not limited to, suggestions to overcome the limitations identified in the previous section.

- The proposed framework for the integration of BIM and construction simulation could benefit greatly by implementing a simulation optimisation method. To this end, an optimisation module capable of evaluating the outputs of the simulation module should be designed. By iterating between the optimisation and simulation modules, a suitable solution for resource allocation and construction alternatives based on project-specific constraints could be automatically detected by the system to facilitate the decision-making process. Such a module could be extended to also find optimal locations of material storage sites.

- While the proposed integrated simulation modelling approach is based on the distributed simulation paradigm in the sense that several models of construction activities are combined together (federated) to form a more complex simulation model (federation), the system at this stage is not prepared to run concurrently on different computers. This aspect describes another meaning for distributed simulation, which has the added benefits of enabling wider collaboration in the model development process, and speeding up the execution time of more complex federations by leveraging parallel (distributed) computing. To this end, developing an HLA-compliant application based on the proposed framework should be considered as a next step. The *distributed simulation component* of the proposed framework should match the specifications of the HLA standard *Run-time Infrastructure* component and provide services to exchange information between federates in different servers (IEEE Computer Society 2010). The capabilities of game engines in regards to developing multi-player games could be leveraged to accomplish such an application.

- A mechanism to consider the continuous motion of resources in the simulation-based animations would provide decision-makers with more insightful information on the modelled system to make an informed decision. These type of animations could contribute to improving health and safety in construction sites by facilitating the process of identifying hazards produced by multiple resources in limited space, for example. As previously discussed, the trace files already provide sufficient data to implement such a solution. Notably, this can be achieved without redefining the discretisation of the federates to an impractical higher granularity, such as modelling every step that a worker takes to move materials.

- Test the framework in a real-life construction project and evaluate its impact on supporting informed decision-making.

7.8 Final remarks

This thesis presented a framework for the integration of BIM and construction simulation by investigating existing modelling and simulation approaches and identifying aspects that enable, firstly, an appropriate capturing of constraints frequently found in construction projects, and secondly, model reuse. The framework enables to semi automatically generate a BIM-based construction simulation model and produces animations based on the simulation results. The developed framework was implemented in a game engine to leverage its capabilities and achieve the proposed functionality. The thesis presented an illustrative case study to demonstrate the feasibility of the framework and its usefulness as a tool to support planning and informed decision-making.

The thesis also addressed the research objectives and met the primary aim of the project, namely, to investigate how to use a BIM model to facilitate the development of a construction simulation model that takes into account constraints related to resource allocation and task interdependencies. Limitations of the research were discussed and future work has been outlined.

References

Abdelkhalek, Sherif and Tarek M Zayed (2020). 'Simulation-based planning of concrete bridge deck inspection with non-destructive technologies'. In: *Automation in Construction* 119, p. 103337.

Abdelmegid, MA et al. (2020). 'Barriers to adopting simulation modelling in construction industry'. In: *Automation in Construction* 111, p. 103046.

AbouRizk, Simaan M (2010). 'Role of simulation in construction engineering and management'. In: *Journal of Construction Engineering and Management* 136.10, pp. 1140–1153.

AbouRizk, Simaan M et al. (2011). 'Research in modeling and simulation for improving construction engineering operations'. In: *Journal of Construction Engineering and Management* 137.10, pp. 843–852.

Akhavian, Reza and Amir H Behzadan (2012). 'An integrated data collection and analysis framework for remote monitoring and planning of construction operations'. In: *Advanced Engineering Informatics* 26.4, pp. 749–761.

Anagnostou, Anastasia and Simon JE Taylor (2017). 'A distributed simulation methodological framework for OR/MS applications'. In: *Simulation Modelling Practice and Theory* 70, pp. 101–119.

Anderson, Eike Falk et al. (2013). 'Choosing the infrastructure for entertainment and serious computer games-a whiteroom benchmark for game engine selection'.

In: *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2013 5th International Conference on*. IEEE, pp. 1–8.

Azhar, Salman (2011). 'Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry'. In: *Leadership and Management in Engineering* 11.3, pp. 241–252.

Bajjou, Mohamed Saad and Anas Chafi (2020). 'Lean construction and simulation for performance improvement: a case study of reinforcement process'. In: *International Journal of Productivity and Performance Management*.

Baniassadi, Farshid, Amin Alvanchi and Ali Mostafavi (2018). 'A simulation-based framework for concurrent safety and productivity improvement in construction projects'. In: *Engineering, Construction and Architectural Management* 25.11, pp. 1501–1515.

Barkokebas, Beda et al. (2017). 'Building information modelling and simulation integration for modular construction manufacturing performance improvement'. In: *Proceedings of the European Modeling and Simulation Symposium, 2017*. Barcelona, Spain, pp. 409–415.

Ben-Alon, Lola and Rafael Sacks (2017). 'Simulating the behavior of trade crews in construction using agents and building information modeling'. In: *Automation in Construction* 74, pp. 12–27.

Bille, Ross et al. (2014). 'Extending building information models into game engines'. In: *Proceedings of the 2014 Conference on Interactive Entertainment*. ACM, pp. 1–8.

Broadbent, Nicholas (2011). *BIM and game engines: How we present 4D models using our design suite and game engine*. URL: `http://aucache.autodesk.com/au2011/sessions/5801/class_handouts/v2_AB5801_Broadbent_Handout.pdf` (visited on 30/10/2019).

Ceylan, Arda and Murat M Gunal (2011). 'A methodology for developing DES models: Event graphs and SharpSim'. In: *Proceedings of the 2011 European Modelling and Simulation. Proceedings of EMSS: The European Modelling & Simulation Symposium 2011*.

Chang, David Y and Md Saiful Hoque (1989). 'A knowledge-based simulation system for construction process planning'. In: *Proceedings of the 6th International Symposium on Automation and Robotics in Construction*, pp. 348–355.

Chang, Soowon et al. (2015). 'BIM-integrated simulation of construction operation simulation for lean production management'. In: *2015 Modular and Offsite Construction [MOC] Summit & 1st International Conference on the Industrialization of Construction [ICIC]*, p. 9.

Chen, Hung-Ming and Pu-Hao Huang (2013). '3D AR-based modeling for discrete-event simulation of transport operations in construction'. In: *Automation in Construction* 33, pp. 123–136.

Chen, Shiwei, Kailun Feng and Weizhuo Lu (2019). 'A Simulation-Based optimisation for contractors in precast concrete projects'. In: *10th Nordic Conference on Construction Economics and Organization*. Emerald Publishing Limited, pp. 137–145.

Conrads, Alena et al. (2017). 'Assessing maintenance strategies for cutting tool replacements in mechanized tunneling using process simulation'. In: *Journal of Simulation* 11.1, pp. 51–61.

Dagkakis, Georgios and Cathal Heavey (2016). 'A review of open source discrete event simulation software for operations research'. In: *Journal of Simulation* 10.3, pp. 193–206.

Dahmann, Judith S, Richard M Fujimoto and Richard M Weatherly (1998). 'The DoD high level architecture: an update'. In: *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*. Vol. 1. IEEE, pp. 797–804.

Dalton, Ben and Maxwell Parfitt (2013). *Immersive visualization of building information models*. Tech. rep. 1.0. Design Innovation Research Center Working Paper.

Dib, Hazar and Nicoletta Adamo-Villani (2013). 'Serious sustainability challenge game to promote teaching and learning of building sustainability'. In: *Journal of Computing in Civil Engineering* 28.5, A4014007.

Du, Jing et al. (2018). 'Zero latency: Real-time synchronization of BIM data in virtual reality for collaborative decision-making'. In: *Automation in Construction* 85, pp. 51–64.

Eastman, Chuck et al. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons.

Ebrahimy, Yasser et al. (2011). 'Simulation modeling and sensitivity analysis of a tunneling construction project's supply chain'. In: *Engineering, Construction and Architectural Management* 18.5, pp. 462–480.

ElNimr, Amr A and Yasser Mohamed (2011). 'Application of gaming engines in simulation driven visualization of construction operations'. In: *Journal of Information Technology in Construction (ITcon)* 16.3, pp. 23–38.

Feng, Kailun et al. (2018). 'An Integrated Environment–Cost–Time Optimisation Method for Construction Contractors Considering Global Warming'. In: *Sustainability* 10.11, p. 4207.

Fu, Feng (2018). 'Design and Analysis of Complex Structures'. In: *Design and Analysis of Tall and Complex Structures*, pp. 177–211.

Furian, Nikolaus et al. (2015). 'A conceptual modeling framework for discrete event simulation using hierarchical control structures'. In: *Simulation modelling practice and theory* 56, pp. 82–96.

Gerek, Ibrahim Halil et al. (2015). 'Modelling masonry crew productivity using two artificial neural network techniques'. In: *Journal of Civil Engineering and Management* 21.2, pp. 231–238.

Golzarpoor, Hamed et al. (2017). 'A non-queue-based paradigm in Discrete-Event-Simulation modelling for construction operations'. In: *Simulation Modelling Practice and Theory* 77, pp. 49–67.

Hajjar, Dany and Simaan M AbouRizk (1999). 'Simphony: an environment for building special purpose construction simulation tools'. In: *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 2*, pp. 998–1006.

Halpin, Daniel W (1977). 'CYCLONE: Method for modeling job site processes'. In: *Journal of the Construction Division* 103.3, pp. 489–499.

Halpin, Daniel W (1990). 'MicroCYCLONE user's manual'. In: *Division of Construction Engineering and Management, Purdue University, West Lafayette, Indiana.*

Hamdan, Samer Bu et al. (2015). 'A BIM-based simulation model for inventory management in panelized construction'. In: *Proceedings of the International Symposium on Automation and Robotics in Construction.* Vol. 32. IAARC Publications, p. 1.

Hassan, Marwa M and Stan Gruber (2008). 'Simulation of concrete paving operations on Interstate-74'. In: *Journal of Construction Engineering and Management* 134.1, pp. 2–9.

Hassanein, A and J Melin (1997). 'Crew design methodology for construction contractors'. In: *Journal of Construction Engineering and Management* 123.3, pp. 203–207.

Hendry, AW (2001). 'Masonry walls: Materials and construction'. In: *Construction and Building Materials* 15.8, pp. 323–330.

Hevner, Alan R et al. (2004). 'Design science in information systems research'. In: *MIS quarterly*, pp. 75–105.

Hou, Lei et al. (2017). 'Discrete firefly algorithm for scaffolding construction scheduling'. In: *Journal of Computing in Civil Engineering* 31.3, p. 04016064.

Huang, Rong-Yau and Daniel W Halpin (1994). 'Visual construction operation simulation: the DISCO approach'. In: *Computer-Aided Civil and Infrastructure Engineering* 9.3, pp. 175–184.

Al-Hussein, Mohamed et al. (2006). 'Integrating 3D visualization and simulation for tower crane operations on construction sites'. In: *Automation in Construction* 15.5, pp. 554–562.

IEEE Computer Society (2010). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)— Framework and Rules.* Tech. rep. IEEE Std. 1516-2010.

Ioannou, PG (1989). 'UM-CYCLONE reference manual'. In: *Technical Report UMCE-89-11.* University of Michigan Michigan.

Jeong, WoonSeong et al. (2016). 'BIM-integrated construction operation simulation for just-in-time production management'. In: *Sustainability* 8.11, p. 1106.

Jin, Haifeng et al. (2020). 'Multiobjective Optimization for Scaffolding Space Planning in Industrial Piping Construction Using Model-Based Simulation Programming'. In: *Journal of Computing in Civil Engineering* 34.1, p. 06019001.

Kalk, Anthony (1980). *INSIGHT: Interactive simulation of construction operations using graphical techniques.* Vol. 238. Stanford University, Department of Civil Engineering.

Kamat, Vineet Rajendra (2003). 'VITASCOPE: Extensible and scalable 3D visualization of simulated construction operations'. Doctoral dissertation. Virginia Tech.

Kamat, Vineet Rajendra and Julio Cesar Martinez (2001). 'Visualizing simulated construction operations in 3D'. In: *Journal of Computing in Civil Engineering* 15.4, pp. 329–337.

Kamat, Vineet Rajendra and Julio Cesar Martinez (2003). 'Validating complex construction simulation models using 3D visualization'. In: *Systems Analysis Modelling Simulation* 43.4, pp. 455–467.

Kasputis, Stephen and Henry C Ng (2000). 'Composable simulations'. In: *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165).* Vol. 2. IEEE, pp. 1577–1584.

Kim, Kyungki, Yong K Cho and Kinam Kim (2018). 'BIM-based decision-making framework for scaffolding planning'. In: *Journal of Management in Engineering* 34.6, p. 04018046.

Kitchenham, Barbara, Lesley Pickard and Shari Lawrence Pfleeger (1995). 'Case studies for method and tool evaluation'. In: *IEEE software* 12.4, pp. 52–62.

König, Markus et al. (2012). 'Intelligent BIM-based construction scheduling using discrete event simulation'. In: *Proceedings of the Winter Simulation Conference.* Winter Simulation Conference, p. 59.

Krantz, Jan et al. (2015). 'Assessing embodied energy and greenhouse gas emissions in infrastructure projects'. In: *Buildings* 5.4, pp. 1156–1170.

Lambert, James (2014). *Making a Game Engine: Core Design Principles.* URL: https://www.gamedev.net/tutorials/programming/general-and-gameplay-programming/making-a-game-engine-core-design-principles-r3210/ (visited on 04/02/2020).

Larsson, Robert and Martin Rudberg (2019). 'Impact of Weather Conditions on In Situ Concrete Wall Operations Using a Simulation-Based Approach'. In: *Journal of Construction Engineering and Management* 145.7, p. 05019009.

Law, Averill M and W David Kelton (2000). *Simulation modeling and analysis.* Vol. 3. McGraw-Hill New York.

Lee, Dong-Eun et al. (2010). 'Integrated simulation system for construction operation and project scheduling'. In: *Journal of Computing in Civil Engineering* 24.6, pp. 557–569.

Lee, Ghang, Rafael Sacks and Charles M Eastman (2006). 'Specifying parametric building object behavior (BOB) for a building information modeling system'. In: *Automation in construction* 15.6, pp. 758–776.

Lee, SangHyun et al. (2013). 'Grand challenges in simulation for the architecture, engineering, construction, and facility management industries'. In: *Computing in Civil Engineering (2013)*, pp. 773–785.

Leite, Fernanda et al. (2016). 'Visualization, information modeling, and simulation: Grand challenges in the construction industry'. In: *Journal of Computing in Civil Engineering* 30.6, p. 04016035.

Li, Shiqi, Yan Jia and Junfeng Wang (2012). 'A discrete-event simulation approach with multiple-comparison procedure for stochastic resource-constrained project scheduling'. In: *The International Journal of Advanced Manufacturing Technology* 63.1-4, pp. 65–76.

Liu, Hexu, Mohamed Al-Hussein and Ming Lu (2015). 'BIM-based integrated approach for detailed construction scheduling under resource constraints'. In: *Automation in Construction* 53, pp. 29–43.

Liu, Liang-Yun (1991). 'COOPS: Construction object-oriented process simulation system.' Doctoral dissertation. University of Michigan.

Lockley, Steve, Claudio Benghi and Martin Cerny (2017). 'Xbim.Essentials: A library for interoperable building information applications'. In: *The Journal of Open Source Software* 2.20, p. 473.

Long, Gavin (2010). 'A detailed investigation of the applicability and utility of simulation and gaming in the teaching of civil engineering students'. Doctoral dissertation. University of Nottingham.

Lu, Ming and Lap-Chi Wong (2007). 'Comparison of two simulation methodologies in modeling construction systems: Manufacturing-oriented PROMODEL vs. construction-oriented SDESA'. In: *Automation in Construction* 16.1, pp. 86–95.

Lu, Weizhuo and Thomas Olofsson (2014). 'Building information modeling and discrete event simulation: Towards an integrated framework'. In: *Automation in Construction* 44, pp. 73–83.

Martinez, Julio Cesar (1996). 'STROBOSCOPE: State and resource based simulation of construction processes'. Doctoral dissertation. University of Michigan.

Martinez, Julio Cesar (2001). 'EZStrobe-general-purpose simulation system based on activity cycle diagrams'. In: *Proceeding of the 2001 Winter Simulation Conference.* Vol. 2. IEEE, pp. 1556–1564.

Martinez, Julio Cesar (2009). 'Methodology for conducting discrete-event simulation studies in construction engineering and management'. In: *Journal of Construction Engineering and Management* 136.1, pp. 3–16.

Natephra, Worawan et al. (2017). ‘Integrating building information modeling and virtual reality development engines for building indoor lighting design’. In: *Visualization in Engineering* 5.1, p. 19.

Nikakhtar, Amin et al. (2011). ‘Comparison of two simulation software for modeling a construction process’. In: *2011 Third International Conference on Computational Intelligence, Modelling & Simulation.* IEEE, pp. 200–205.

Odeh, AM (1992). ‘Construction integrated planning and simulation model’. Doctoral dissertation. University of Michigan.

Osorio-Sandoval, Carlos Arturo (2015). ‘Determinación del impacto del ausentismo de la mano de obra sobre el tiempo de ejecución de las actividades de construcción de vivienda’. Master’s thesis. Universidad Autónoma de Yucatán.

Osorio-Sandoval, Carlos Arturo, Christian Koch and Walid Tizani (2017). ‘Computer technology for serious games in education: A literature review’. In: *Proceedings of the 24th EG-ICE International Workshop on Intelligent Computing in Engineering 2017.* EG-ICE, pp. 239–249.

Osorio-Sandoval, Carlos Arturo, Walid Tizani and Christian Koch (2018). ‘A method for discrete event simulation and building information modelling integration using a game engine’. In: *Advances in Computational Design* 3.4, pp. 405–418.

Osorio-Sandoval, Carlos Arturo et al. (2016). ‘Discrete Event Simulation Analysis of the Effect of Labor Absenteeism on the Duration of Construction Activities in Housing Projects’. In: *International Journal of Engineering Research and Applications* 6.11, pp. 46–55.

Pereira, Estacio, SangUk Han and Simaan M AbouRizk (2018). ‘Integrating Case-Based Reasoning and Simulation Modeling for Testing Strategies to Control Safety Performance'. In: *Journal of Computing in Civil Engineering* 32.6, p. 04018047.

Petridis, Panagiotis et al. (2010). 'An engine selection methodology for high fidelity serious games'. In: *2010 Second International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*. IEEE, pp. 27–34.

Project Management Institute (2013). *PMBOK® Guide: A Guide to the Project Management Body of Knowledge*. 5th ed. Project Management Institute.

Puri, Vivek and Julio Cesar Martinez (2013). 'Modeling of simultaneously continuous and stochastic construction activities for simulation'. In: *Journal of construction engineering and management* 139.8, pp. 1037–1045.

Rahm, Tobias et al. (2016). 'Evaluation of disturbances in mechanized tunneling using process simulation'. In: *Computer-Aided Civil and Infrastructure Engineering* 31.3, pp. 176–192.

Rekapalli, Prasant V and Julio Cesar Martinez (2010). 'Discrete-event simulation-based virtual reality environments for construction operations: Technology introduction'. In: *Journal of Construction Engineering and Management* 137.3, pp. 214–224.

Robinson, Stewart et al. (2004). 'Simulation model reuse: definitions, benefits and obstacles'. In: *Simulation modelling practice and theory* 12.7-8, pp. 479–494.

Ruegg, Christoph, Marcus Cuda and Jurgen Van Gael (2019). *Math.Net Numerics*. Version 4.8.0. URL: https://numerics.mathdotnet.com/.

Sargent, Robert G (2013). 'Verification and validation of simulation models'. In: *Journal of simulation* 7.1, pp. 12–24.

Schaufelberger, John E and Giovanni C Migliaccio (2019). *Construction equipment management*. Routledge.

Seo, JoonOh, SangHyun Lee and Jongwon Seo (2016). 'Simulation-based assessment of workers' muscle fatigue and its impact on construction operations'. In: *Journal of Construction Engineering and Management* 142.11, p. 04016063.

Shi, Jonathan Jingsheng (1999). 'Activity-based construction (ABC) modeling and simulation method'. In: *Journal of construction engineering and management* 125.5, pp. 354–360.

Song, Lingguang and Neil N Eldin (2012). 'Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling'. In: *Automation in Construction* 27, pp. 32–39.

Taghaddos, Hosein (2010). 'Developing a generic resource allocation framework for construction simulation'. Doctoral dissertation. University of Alberta.

The National BIM Report 2018 (2019). URL: https://www.thenbs.com/knowledge/the-national-bim-report-2018 (visited on 17/05/2019).

Turner, Christopher J et al. (2016). 'Discrete event simulation and virtual reality use in industry: New opportunities and future trends'. In: *IEEE Transactions on Human-Machine Systems* 46.6, pp. 882–894.

Unity Technologies (2017). *Unity Asset Store.* URL: https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2017-3-32351.

Unity Technologies (2019). *Unity - Manual: Unity User Manual (2019.1).* URL: https://docs.unity3d.com/Manual/UnityManual.html.

Varol, Ali Emre, Arda Ceylan and Murat M. Gunal (2011). *SharpSim tutorials.* 33 pp.

Wang, Haidong and Zhiqiang John Zhai (2016). 'Advances in building simulation and computational techniques: A review between 1987 and 2014'. In: *Energy and Buildings* 128, pp. 319–335.

Wang, Wei-Chih et al. (2014). 'Integrating building information models with construction process simulations for project scheduling support'. In: *Automation in Construction* 37, pp. 68–80.

Werner, Michael and Simaan M AbouRizk (2015). 'Simulation case study: Modelling distinct breakdown events for a tunnel boring machine excavation'. In: *Proceedings of the 2015 Winter Simulation Conference*. IEEE Press, pp. 3234–3245.

Wu, Chengke, Rui Jiang and Xiao Li (2016). 'Integration of BIM and Computer Simulations in Modular Construction, A Case Study'. In: *Modular and Offsite Construction (MOC) Summit Proceedings*, pp. 59–66.

Wu, I-Chen et al. (2010). 'Bridge construction schedule generation with pattern-based construction methods and constraint-based simulation'. In: *Advanced Engineering Informatics* 24.4, pp. 379–388.

Wu, Wei and Ishan Kaushik (2015). 'A BIM-based educational gaming prototype for undergraduate research and education in design for sustainable aging'. In: *Winter Simulation Conference (WSC), 2015*. IEEE, pp. 1091–1102.

Younes, Taha, Frank Mi-Way Ni and Susan Tighe (2020). 'Risk analysis in paving operations using discrete event simulation: a case study of Taiwan permeable asphalt concrete pavement pilot road project'. In: *International Journal of Pavement Engineering* 21.7, pp. 830–840.

Zayed, Tarek M and Daniel W Halpin (2001). 'Simulation of concrete batch plant production'. In: *Journal of Construction Engineering and Management* 127.2, pp. 132–141.

Zhang, Cheng, Tarek M Zayed and Amin Hammad (2008). 'Resource management of bridge deck rehabilitation: Jacques Cartier bridge case study'. In: *Journal of Construction Engineering and Management* 134.5, pp. 311–319.

Zhang, Peiyao et al. (2019). 'An agent-based modeling approach for understanding the effect of worker-management interactions on construction workers' safety-related behaviors'. In: *Automation in construction* 97, pp. 29–43.

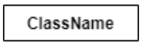
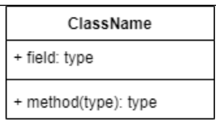
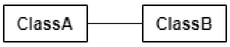
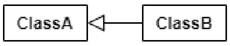
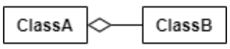
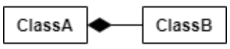
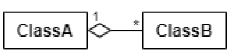
Zhang, Sijie et al. (2013). 'Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules'. In: *Automation in Construction* 29, pp. 183–195.

Appendix A

UML notation

Table A.1 depicts the notation used in the UML diagrams in this document.

Table A.1: UML notation

Name	Symbol	Description
Class		Represents a class without including its fields or methods
Class with fields and methods		Represents a class including its fields and methods
Association		There is a simple association between ClassA and ClassB
Inheritance		Represents an "is a" relationship. ClassB is a specialization of ClassA and inherits its fields and methods
Aggregation		Represents a "part of" relationship. ClassB is part of ClassA, but they have separate lifetimes
Composition		A special type of aggregation in which the parts cannot stand by themselves
Dependency		ClassA depends on ClassB. Changes on ClassB affect ClassA
Multiplicity		How many instances of each class take part in the relationship