A BIM-INTEGRATED AGENT-BASED SIMULATION METHOD FOR TIME-SPACE CONFLICT DETECTION AMONG MOBILE RESOURCES IN CONSTRUCTION PROJECTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAMAN KHATAEI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CIVIL ENGINEERING

SEPTEMBER 2024

Approval of the thesis:

A BIM-INTEGRATED AGENT-BASED SIMULATION METHOD FOR TIME-SPACE CONFLICT DETECTION AMONG MOBILE RESOURCES IN CONSTRUCTION PROJECTS

submitted by **SAMAN KHATAEI** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** i**n Civil Engineering, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, **Graduate School of Natural and Applied Sciences** _____

Prof. Dr. Erdem Canbay
Head of the Department, **Civil Engineering** _____

Assist. Prof. Dr. Aslı Akçamete Güngör
Supervisor, **Civil Engineering, METU** _____

Prof. Dr. Rifat Sönmez
Co-Supervisor, **Civil Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Güzide Atasoy Özcan
Civil Engineering, METU _____

Assist. Prof. Dr. Aslı Akçamete Güngör
Civil Engineering, METU _____

Assoc. Prof. Dr. Hande Işık Öztürk
Civil Engineering, METU _____

Assist. Prof. Dr. Saman Aminbakhsh
Civil Engineering, Atılım University _____

Assist. Prof. Dr. Gözde Bilgin
Civil Engineering, Başkent University _____

Date: 05.09.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Saman Khataei

Signature :

ABSTRACT

A BIM-INTEGRATED AGENT-BASED SIMULATION METHOD FOR TIME-SPACE CONFLICT DETECTION AMONG MOBILE RESOURCES IN CONSTRUCTION PROJECTS

Khataei, Saman
Doctor of Philosophy, Civil Engineering
Supervisor: Assist. Prof. Dr. Aslı Akçamete Güngör
Co-Supervisor: Prof. Dr. Rifat Sönmez

September 2024, 132 pages

Schedule compression techniques are frequently used to meet the construction projects' temporal deadlines. Crashing is a common technique in this regard and requires an increase in the number of resources of the critical path activities to shorten the duration of work. However, any increase in the project resources will increase the interactions among them, leading to time-space conflicts that should be taken into account during crashing analysis. Conventionally, the crashing analysis is performed subjectively, relying on the project planner’s insights or previous project records, which may not lead to reliable estimations. To achieve a more accurate estimation for crashing alternatives, space congestion among the resources should be quantified and involved in the analysis with a project-based and quantitative approach. This study presents a novel method for generating and evaluating the crashing alternatives that are laid on the agent-based simulation (ABS) and BIM model. The method aims to increase the construction simulation fidelity and fulfill some of the current gaps by quantifying time-space conflicts to support the time-cost trade-off and corresponding risk analyses. The method takes a 4D BIM model, an

agent pool, and crashing alternatives as input sets and generates a simulation, in which the agents represent project manpower and machinery resources (crews) that can act independently on the basis of predefined behavioral patterns and move according to an artificial intelligence (AI) pathfinding algorithm supported by their perception of the environment and other agents. Leveraging the independent nature of the agents, a conflict avoidance mechanism that models the crews' behavior in real cases is also developed and integrated with them. In order to evaluate the proposed method, it was implemented in two case examples with distinct environments including a game engine and a commercial simulation tool. The analysis outcomes demonstrated the impact of the time-space conflicts on the crew productivity and provided crew size–project duration estimation as the quantified crashing modes. The method is expected to ameliorate the accuracy of the crashing analysis and help planners anticipate the emergent outcomes related to the different time-cost scenarios with associated risks.

Keywords: Time-Space Conflicts, Agent-based Simulation, BIM, Time-Cost Trade-Off, Optimal Crew Size

ÖZ

İNŞAAT PROJELERİNDE MOBİL KAYNAKLAR ARASINDAKİ ZAMAN-ALAN ÇATIŞMASINI TESPİT İÇİN YAPI BİLGİ MODELLEMESİ VE AJAN TABANLI BİR SİMÜLASYON YÖNTEMİ

Khataei, Saman
Doktora, İnşaat Mühendisliği
Tez Yöneticisi: Dr. Öğr. Üyesi. Aslı Akçamete Güngör
Ortak Tez Yöneticisi: Prof. Dr. Rifat Sönmez

Eylül 2024, 132 sayfa

Proje hızlandırma teknikleri, inşaat projelerinin zamanında tamamlanabilmesi için sıkça kullanılır. Bu bağlamda yaygın olarak kullanılan tekniklerden biri de süre kısaltma (crashing) yöntemidir ve bu yöntem, kritik yol aktivitelerine ait kaynakların artırılarak işin süresinin kısaltılmasını gerektirir. Ancak, proje kaynaklarındaki herhangi bir artış, kaynaklar arasındaki etkileşimleri artırarak zaman-alan çatışmalarına yol açar ve bu durum, süre kısaltma analizinde dikkate alınmalıdır. Geleneksel olarak, süre kısaltma analizi, proje yöneticisinin içgörüleri veya önceki proje kayıtlarına dayanarak öznel bir şekilde gerçekleştirilir, bu da güvenilir tahminlere yol açmayabilir. Süre kısaltma alternatifleri için daha doğru bir tahmin elde edebilmek adına kaynaklar arasındaki alan sıkışıklığı nicel olarak ölçülmeli ve proje bazlı, sayısal bir yaklaşımla analize dahil edilmelidir. Bu çalışmada, süre kısaltma alternatiflerini oluşturmak ve değerlendirmek için zaman-alan çatışmalarını nicel olarak ölçmek üzere ajan tabanlı simülasyon (ABS) ve Yapı Bilgi Modellemesi (YBM) bazlı yeni bir simülasyon yöntemi sunar. Bu yöntem, inşaat simülasyonunun doğruluğunu artırmayı ve zaman-maliyet denklemi ve ilgili risk analizlerini

desteklemek için zaman-alan çatışmalarını ölçerek mevcut eksiklikleri gidermeyi amaçlamaktadır.

Bu metot, 4B YBM modelini, bir ajan havuzunu ve süre kısaltma alternatiflerini girdi olarak alır ve ajanların, proje iş gücünü temsil ettiği bir simülasyon üretir. Bu ajanlar, önceden tanımlanmış davranış kalıplarına göre bağımsız olarak aksiyon alabilirler ve çevre algıları ile diğer ajanlarla etkileşimlerine dayalı olarak bir yapay zeka (AI) yol bulma algoritması aracılığıyla hareket ederler. Ajanların bağımsız doğasından yararlanarak, ekiplerin gerçek durumlarda sergiledikleri davranışları modelleyen bir çatışma önleme mekanizması da geliştirilmiş ve ajanlarla entegre edilmiştir. Önerilen yöntemi değerlendirmek amacıyla, farklı ortamlara sahip iki örnek proje üzerinde, bir oyun motoru ve ticari bir simülasyon aracı kullanılarak uygulanmıştır. Analiz sonuçları, zaman-alan çatışmalarının ekip verimliliği üzerindeki etkisini göstermiş ve ekip sayısı-proje süresi tahmini gibi nicel süre kısaltma modları sağlamıştır. Bu yöntemin, süre kısaltma analizinin doğruluğunu artırması ve planlayıcıların farklı zaman-maliyet senaryolarıyla ilişkili riskleri içeren olası sonuçları öngörmelerine yardımcı olması beklenmektedir.

Anahtar Kelimeler: Zaman-Alan Çatışmaları, Ajan-tabanlı Simülasyon, YBM, Zaman-Maliyet Denklemi, Optimal Ekip Sayısı

In dedication to my wife and my mother, father, and sister with love

ACKNOWLEDGMENTS

I am deeply indebted to many people for their support and guidance throughout my Ph.D. journey. First and foremost, I wish to express my sincere gratitude to my supervisor, Assist. Dr. Aslı Akçamete Güngör. Your insightful feedback, unwavering support, and exceptional mentorship have been invaluable in shaping this research. I am equally grateful to my co-supervisor, Prof. Dr. Rifat Sönmez, for your constant encouragement, critical insights, and expert guidance. I extend my heartfelt thanks to the professors at METU, whose profound knowledge and commitment to excellence have greatly enriched my academic experience. Your collective wisdom and encouragement have been pivotal in navigating the challenges of this academic pursuit.

I owe an immense debt of gratitude to my family. To my beloved wife, Saba Amini, your unwavering love, patience, and endless support have been my greatest source of strength. Your belief in me has kept me going even in the toughest of times. To my mother, Zhale Barhagh, father, Behrouz Khataei, and sister Aysa Khataei, your sacrifices, love, and encouragement have been the bedrock of my achievements. Your constant prayers and unwavering support have been my guiding light. This thesis is as much yours as it is mine. Thank you for believing in me and for your endless encouragement.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xv

CHAPTERS

1 INTRODUCTION 1

1.1 Problem Statement and Motivation 1

1.2 Research Questions 4

1.3 Research Contribution 5

2 LITERATURE REVIEW 7

2.1 Modeling and Simulation 7

2.2 Simulation Paradigms 8

2.2.1 Discrete Event Simulation (DES) 9

2.2.2 Agent-Based Simulation (ABS) 11

2.2.3 System Dynamics (SD) 13

2.3 Anylogic as a Commercial Simulation Tool 14

2.4 Game Engines 16

2.5 Deterministic and Stochastic Approaches 17

2.6 Workspace in Construction 18

2.6.1 Classification of Workspaces 19

2.7 Influence of Time-Space Conflicts on Productivity 20

2.8 Use of Simulations in Construction 22
2.8.1 Process-based Simulations 22
2.8.2 Visual-based Simulations 26
2.9 Project Delay Management 37
2.9.1 Project Acceleration Methods 37
2.10 Time-space Conflict Detection by Simulation Techniques 38
3 BIM-INTEGRATED ABS METHOD FOR QUANTIFYING TIME-SPACE CONFLICTS 49
3.1 Method Architecture 49
3.1.1 Data Extraction, Agent Pool, and Inputs 49
3.1.2 System Configuration 51
3.1.3 Simulation Run Process 54
3.1.4 Simulation Output 59
4 CASE STUDIES 61
4.1 Method Implementation in Gaming Engine 61
4.1.1 BIM Model Creation and Exporting to Game Engine 62
4.1.2 Collision Detection Mechanism in Game Engine 64
4.1.3 Agent State and Data Collection 65
4.1.4 Simulation Run 69
4.1.5 Discussion 78
4.1.6 Approach Limitations 81
4.2 Method Implementation in a Commercial Tool (Anylogic) 81
4.2.1 Data Collection 82
4.2.2 Agent Statechart 83

4.2.3 The Simulation Environment 92
4.2.4 The Simulation Run 93
4.2.5 Results and Discussion 97
4.2.6 Approach Benefits and Limitations 101
5 VERIFICATION & VALIDATION (V&V) 104
5.1 Verification 105
5.2 Validation 108
6 CONCLUSION 110
REFERENCES 115
APPENDICES
A. Case One Simulation Run Screenshots 125
B. Case Two Simulation Run Screenshots 128
CURRICULUM VITAE 131

LIST OF TABLES

TABLES

Table 2.1 Classification of workspaces according to the previous studies (Messi et al., 2022) 20
Table 2.2 A list of the selected VLPs and their use cases 31
Table 2.3 A comparison between the previous and present studies 48
Table 3.1 Calculating M1 & M2 for a sample project 57
Table 4.1 Different duration percentiles and direct costs for different crew sizes .. 78
Table 4.2 Crashing modes for various crew sizes 80
Table 4.3 The system configuration for simulation runs 81
Table 4.3 Crashing analysis considering the three modes 100
Table 5.1 Video sources for V&V 105
Table 5.2 Productivity and duration calculations based on the RSMeans database 109

LIST OF FIGURES

FIGURES

Figure 2.1. A sample of DES implementation 10
Figure 2.2. A CYCLONE model for earthworks (S. M. AbouRizk & Halpin, 1990) 11
Figure 2.3. An SIR model in system dynamics 13
Figure 2.4. Hybrid simulation options using Anylogic (Borshchev, 2014) 15
Figure 2.5. The architecture of Anylogic (Borshchev et al., 2002) 16
Figure 2.6. The simulation environment to model MiC transport logistics (Attajer & Mecheri, 2024) 25
Figure 2.7. The SD model developed for the optimization of policies for PB (Luo, 2024) 26
Figure 2.8. A first overview of BDM elements suggested by Charles Eastman (Eastman, 1975) 28
Figure 2.9. An example schema of IFC (Mendes Da Luz Patacas et al., 2014) 28
Figure 2.10. The integration of the gaming engine with architectural design (Au, 2024) 29
Figure 2.11. A VLP user interface with the node library (left) and workspace (right) 31
Figure 2.12. A 4D BIM-based simulation scene 33
Figure 2.13. A snapshot of the developed simulation (Kamat & Martinez, 2001) . 35
Figure 2.14. A scene of the generated simulation (Al-Hussein et al., 2006) 36
Figure 2.15. Examples of TCTP (left) vs DTCTP (right) 39
Figure 2.16. A simulation scene of the proposed simulation (Ben-Alon & Sacks, 2017) 43
Figure 2.17. Predefined execution patterns (Mirzaei et al., 2018) 45
Figure 2.18. Three bounding volume methods for space conflict analysis (Dashti et al., 2021) 47
Figure 2.19. The DES developed for earthwork processes (Fathi et al., 2023) 47

Figure 3.1. The proposed method architecture 50
Figure 3.2. A pseudocode for extracting BIM model data and inserting to the database table 51
Figure 3.3. The database structure to store model element information 52
Figure 3.4. The pseudocode of A* algorithm (Cui & Shi, 2010) 53
Figure 3.5. An example conflict matrix implemented in Anylogic 54
Figure 3.6. Nominating next destination points for $M1$ calculation 56
Figure 3.7. Calculating other agents' locations to obtain $M2$ 56
Figure 3.8. Determining the overlapping and required areas during a time-space conflict 58
Figure 4.1. BIM-based model of the warehouse project 62
Figure 4.2. Imported model in the game engine environment 63
Figure 4.3. NavMesh mechanism in Unity (adapted from Unity documentation) 63
Figure 4.4. Collision detection script in C# 64
Figure 4.5. Clearance of the collision detection function 65
Figure 4.6. Mobile crane agent state chart 66
Figure 4.7. Extracting BIM model data using Dynamo VLP 67
Figure 4.8. A Python code to insert the data into the database table 68
Figure 4.9. SQLite database filled with the BIM model data 68
Figure 4.10. User interface (UI) for getting user inputs in Unity 69
Figure 4.11. Implementation of multicriteria function in C# 72
Figure 4.12. A simulation scene in Unity 73
Figure 4.13. An example Coroutine implementation in Unity (adapted from Unity manual) 73
Figure 4.14. An implementation of Coroutine in the simulation 74
Figure 4.15. C# script for retrieving data from database 75
Figure 4.16. Task execution duration with different crew sizes with deterministic parameters 76
Figure 4.17. Task execution duration reduction by increasing crew size with deterministic parameters 76

Figure 4.18. Stacked chart of the agent state during the simulation runs for the deterministic approach 77
Figure 4.19. ECDF chart created by the simulation for different crew sizes..... 77
Figure 4.20. State proportion chart regarding an agent interaction in different simulation runs 79
Figure 4.21. Crashing analysis to select the best crashing mode 80
Figure 4.22. The built-in database in Anylogic with BIM model data and baseline schedule 83
Figure 4.23. Simple state (right) and composite state (left) 84
Figure 4.24. Using history state to remember the last visited state 85
Figure 4.25. Inheritance in Java programming language 86
Figure 4.26. The worker agent's environment 86
Figure 4.27. The excavator agent environment with statechart 87
Figure 4.28. The script triggered by taking the idle state 88
Figure 4.29. The agent environment for the excavator agent 88
Figure 4.30. Calculating *r* factor using Java code in Anylogic 89
Figure 4.31. Finding destination point in Java code 90
Figure 4.32. The statechart of the concrete crew agent 91
Figure 4.33. The masonry agent statechart 91
Figure 4.34. BIM model plan view as the simulation environment 92
Figure 4.35. The schedule block for daily work assignments 93
Figure 4.36. The case 2 project model in Autodesk Revit 94
Figure 4.37. The revised schedule with simulation-based start and finish dates 95
Figure 4.38. A simulation run scene in Anylogic 95
Figure 4.39. The list of the tasks for the baseline schedule plan 96
Figure 4.40. The baseline schedule plan Gantt chart 97
Figure 4.41. Estimated schedule plan (grey) vs simulation-based (light blue) schedule plan for crashing mode 1 98
Figure 4.42. Estimated schedule plan (grey) vs simulation-based (light blue) schedule plan for crashing mode 2 99

Figure 4.43. Estimated schedule plan (grey) vs simulation-based (light blue) schedule plan for crashing mode 3 100
Figure 4.44. Crashing analysis to select the best crashing mode 101
Figure 4.45. Agent-level task duration against increasing the crew size 102
Figure 4.46. The gap between the baseline and simulation-based schedule plans by increasing the crew size 102
Figure 5.1. The model adapted from the video resource (Code I) 106
Figure 5.2. The behavior and sequence verification of the agent in comparison with the video frames in comparison with the modeled simulation - Code I 106
Figure 5.3. The sequence verification of the agent in comparison with the video frames - Code II 107
Figure 5.4. The sequence verification of the agent in comparison with the video frames - Code III and IV 107
Figure 5.5. The behavior and sequence verification of the agent in comparison with the video frames in comparison with the modeled simulation - Code V 108
Figure 5.6. The estimated data retrieved from the RSMeans database 109
Figure 6.1. A simulation run scene in Unity environment 125
Figure 6.2. A scene of a simulation run in Unity indicating two agents mounting the steel elements 126
Figure 6.3. A simulation run scene in Unity with more than two agents 127
Figure 6.4. The excavator agent development environment in Anylogic 128
Figure 6.5. The concrete agent development environment in Anylogic 129
Figure 6.6. The simulation landing page environment for adjusting the conflict matrix in Anylogic 129
Figure 6.7. A view of the used database schema in Anylogic 130

CHAPTER 1

INTRODUCTION

This chapter includes the research motivation concerning the current gaps and the proposed solution roadmap details. Moreover, the research questions to outline the contribution of the study are included in the current chapter.

1.1 Problem Statement and Motivation

Time and budget are the determinative factors in any construction project, as they directly impact the project's success and sustainability. Efficient management of the project resources ensures the profit for the contractor side and the quality of the asset for the owner side. Delays and cost overruns lead to financial loss, reputational damage, and negative environmental effects. In general, the budget and timeline for each project depend heavily on the accuracy of the initial estimations, which becomes even more critical when decisions are made to expedite projects in order to compensate for delays and meet deadlines. In this regard, working space should be considered as a limited project resource amid the project acceleration planning.

Schedule crashing is one of the common methods for project acceleration meaning an increase in the project resources to improve the productivity at the project level. In this method, more resources are assigned to the activities located on the critical path, thereby reducing the overall project duration. However, any increase in the project resources creates space conflicts which are followed by declination in activity level productivity. Therefore, a trade-off is essential to be performed between the increased resources and achieved productivity growth as crashing alternatives which result in Discrete Time-Cost Trade-Off (DTCTO) analyses. This process is

traditionally handled by the project planner with a subjective approach grounded on previous experiences. Hence, a quantified method to offer crashing alternatives contributes to the estimation reliability and decision-making process, especially in delayed projects.

While space conflicts commonly arise during the construction phase, traditional methods often fail to quantify them beforehand, typically identifying them only after they've occurred (Semenov et al., 2014). This delay can lead to further setbacks and increased expenses (Chua et al., 2010; Hosny et al., 2020). There has been a growing interest in literature to incorporate spatial considerations into planning, aiming to detect time-space conflicts early on.

Each construction crew during the construction stage requires adequate workspace to perform the assigned task with the expected productivity and under safety considerations. The space conflicts among the crews cause interruptions in the tasks' execution followed by productivity reduction or constructability issues. This effect extends the associated activity duration which is reflected in the overall project duration. Therefore, taking the working space as a critical factor in planning construction activities is essential. However, the dynamic nature of the needed workspace and various space amount requirements regarding crew type make it challenging to achieve reliable estimations over space planning. Involving space in the project schedule plan demands a deep perception of the geometric surroundings of the elements in the associated task with safety considerations.

Simulation is a method to mimic complex systems with adequate fidelity to detect emergent system behaviors that are not noticeable with ordinary analyses. As construction projects are considered complex systems, by using simulation techniques, space congestion can be detected taking into account the complex interaction of the crews and the dynamic nature of the project. There are various simulation types designed to meet different requirements, thus, there should be a detailed analysis of the problem aspects and the expected output to end up with an appropriate technique.

Agent-based simulation (ABS) is one of the simulation types designed to model systems that include independent parts having a determinant effect on the system behavior. ABS consists of the agents owning distinct behavior patterns to make intelligent decisions upon the perception of the environment or other agents. This approach fits well with construction projects with the crews interacting with each other and the environment. Each crew can be modeled as an agent with specific tasks, resources, and decision-making rules, allowing the simulation to capture construction activities' complex and often unpredictable nature. The required workspace for each agent with a set of virtual sensory systems should be modeled to implement the ABS in the construction space conflict detection problem. Furthermore, a conflict avoidance mechanism should be included in the model to reflect the actions taken by the crews in order to minimize the negative effects of congestion either by changing the sequence of the tasks or giving priority to the critical activities. It is aimed that the mentioned aspects provide higher fidelity to the construction workflow simulation, thus, narrowing the existing literature gaps in the space conflict problem. To include the mentioned aspects in the simulation, an object-oriented 3D model of the project is required in the first place.

With the advent of Building Information Modeling (BIM), the integration of time dimensions into 3D objects to form 4D BIM models has become feasible, leading to the development of new approaches to detect and mitigate space conflicts (Bansal, 2011; Khataei et al., 2020; Tak et al., 2021; J. P. Zhang & Hu, 2011). However, existing methods still have gaps that need addressing, primarily due to oversimplified representations of workspaces at various levels of abstraction. These simplifications fail to capture the complexities of real project conditions, including conflict dynamics and crew movement patterns (Riley & Sanvido, 1995). Consequently, the disparity between model representation and on-site conditions widens. Moreover, previous approaches often impose predefined and fixed behaviors on simulation entities, neglecting their ability to perceive and make decisions intelligently. A top-down approach to this issue falls short of providing a reliable and detailed model of the dynamic and highly interactive construction environment. Moreover,

uncertainties surrounding model parameters should also be considered (Ahuja & Nandakumar, 1985). Assuming deterministically for congestion analysis lacks reliability and accuracy, as any change in project conditions could invalidate the analysis. Finally, the complexities of each construction project demand a unique analysis of time-space conflicts. Even projects that appear similar can yield disparate outcomes when faced with concurrent increases in resources.

The proposed method enriches DTCTO analyses in the crashing process by dynamically incorporating the spatial dimension through a bottom-up simulation approach. Unlike conventional methodologies, this analytical approach enhances the alignment of schedules with real-world conditions, thus yielding a more dependable estimation and nuanced understanding of how space constraints impact resource allocation and activity productivity. This approach allows the decision-makers to achieve the most viable crashing alternatives by anticipating inefficiencies and potential conflicts through resource increases.

1.2 Research Questions

The current research aims to find answers to the following questions in order to fill the gaps mentioned.

1. How does BIM-based ABS simulation enhance the accuracy and reliability of construction simulation for supporting crashing analysis?

2. How does the proposed approach integrate independent behavior and movement of project crews within an uncertain 3D environment? and How does the inclusion of uncertainties affect the robustness and versatility of simulation outcomes compared to deterministic simulations?

3. How does the integration of the new approach support crashing analysis for optimal crew size selection in construction projects?

4. What are the potential future enhancements of the method that can contribute to the crashing analysis?

1.3 Research Contribution

In the proposed approach, BIM-based models not only serve as the backbone of the simulation environment but also offer vital simulation parameters derived from the data of model objects. These parameters include geometric positions, connections, quantities, weights, and materials, among others. Beyond providing a three-dimensional visual representation of construction activities for methodological evaluation, our approach diverges from deterministic 4D simulations by adapting uncertainties. This is achieved through the integration of weighted number generators based on probabilistic values and distributions throughout the simulation process. Users have the flexibility to configure these probabilities based on their preferences, allowing for a customizable or deterministic approach.

The proposed methodology aligns with the principles of ABS within a BIM-based model environment. Here, crews or equipment are depicted as agents endowed with unique properties and behaviors, operating autonomously and interacting in real time. Each agent type owns a distinct set of behaviors tailored to the crew workflow in real projects such as the required workspace, the patterns of movement, supply point need, etc. These agents are driven by specific tasks, possess environmental awareness, and can adapt their behavior in response to changes in their surroundings. Furthermore, a tailored artificial intelligence (AI) algorithm to facilitate pathfinding and obstacle avoidance for these agents is implemented.

Furthermore, the severity of space congestion is determined concerning the geometrically congested space compared to the whole workspace which is followed by the dynamic productivity revision of the agent. Moreover, the space conflicts that lead to constructibility issues are determined using the conflict matrix. Therefore, in

a conflict case, the agent initially checks the matrix for constructability issues, then the productivity revision.

The present study combines 4D BIM and ABS to model and simulate the activities and their assigned resources. 4D BIM offers a comprehensive simulation environment by integrating a schedule with a 3D object-oriented framework and detailed data. It provides crucial data for each simulation run, including postural data, materials, and quantities, among other necessary details. Moreover, ABS contributes to the quantification of the time-space conflicts in several aspects. Firstly, it models crews acting independently based on unique behavioral patterns, allowing for the realistic imitation of crew movements during task execution, including the paths connecting various execution points. Secondly, it incorporates uncertainties in construction work using a probabilistic approach to simulation parameters, enabling the detection and consideration of emergent and unpredictable cases in the analysis. Thirdly, it simulates crew-crew and crew-environment interactions by creating agents capable of sensing dynamic environmental changes and the states of other agents. Finally, it provides a mechanism to model conflict avoidance behavior in real-world conditions, where project crews sharing the same zone adjust their execution patterns to avoid spatial congestion. To implement the mechanism, each agent in ABS employs an integrated multicriteria function to make appropriate decisions accordingly. To sum up, the current method aims to increase the fidelity of modeling the activities and corresponding crews in construction simulation by detecting time-space conflicts to help the project planners perform crashing analyses, assess time-cost risks, and determine optimal crew size with a more reliable method by overcoming some of the current limitations.

CHAPTER 2

LITERATURE REVIEW

This chapter outlines the previous works conducted for time-space conflict detection and the evaluation of the associated effects on the construction workflow. Following the previous studies the latest progress regarding the solution to time-space conflict detection is analyzed to appraise the level of the study contribution in the literature.

2.1 Modeling and Simulation

Modeling can be defined as crafting a representation that reflects the structure and functionality of a particular system and is a simplified version of the actual system. The main aim of developing a model is to allow analysts to forecast the outcomes of alterations made to the system. Ideally, a model should accurately replicate the authentic system, capturing its essential traits. While a model is an abstraction and cannot encompass every facet of reality, it enables us to interact with the world in a streamlined way, sidestepping the complexities, risks, and irreversible aspects of actuality. However, as the complexity of modeling increases, it also raises the analysis costs. Therefore, a balance must be struck between model accuracy and affordability to achieve an acceptable and cost-effective solution. In general, the models generated for the simulation purpose are mathematical models (Maria, 1997).

Various modeling tools, languages, and methods, such as mathematical, process-based, economical, etc., are selected depending on the requirements of the system being abstracted (Naciri et al., 2024). Models can range from simplified forms like a wooden sword, a chess table, or the German Kriegsspiel, to more intricate and advanced representations such as computer models, digital prototypes, or

comprehensive analytical frameworks. The choice of model complexity and type is determined by the specific characteristics, objectives, and technological capabilities.

A simulation is tracing the effect of state changes on the developed model by a single or repeated run. A simulation can be considered a risk-free environment where unpredictable and emergent outcomes can be detected and analyzed. Many studies underscore the key role of simulation in optimizing investments, improving production layouts, and streamlining manufacturing systems. By constantly refining processes through simulation, companies can enhance decision-making and boost performance. This modern approach has outpaced the outdated method of using physical mock-ups in traditional prototyping, which often led to bottlenecks.

To set up a simulation experiment, the system components should be outlined, including inputs, functional connections, and performance metrics (Maria, 1997). To start with the development, the aspect of the system to be analyzed and the expected functionality and objectives should be identified first. This phase determines the level of simulation fidelity, a factor indicating the detailing level of which the model inherits from the real system. As fidelity increases, while accuracy improves, complexity also grows. Thus, a careful balance between fidelity and complexity is essential. In general, aiming to obtain an understandable, extendable, and reusable simulation, it is recommended to avoid unnecessary complexities in the model (Robinson, 2005).

2.2 Simulation Paradigms

Simulation paradigms (types) can be grouped based on various influencing factors. For example, time-related distinctions classify simulations as either static or dynamic. Certainty considerations differentiate simulations into stochastic, which incorporate uncertainties via probability distributions, and deterministic types. Another notable distinction is between Agent-based simulation (ABS) or model (ABM) and equation-based simulation (EBS or EBM). ABS, a more contemporary

approach, features autonomous agents capable of decision-making based on interactions with other agents or the environment. Conversely, EBS relies on equations evaluated during simulation runs without agent interactions (Parunak, 1998). Additionally, simulations can be categorized as Continuous Simulation (CS), or Discrete Event Simulation (DES) based on variable changes over time intervals. DES maintains constant variable values within intervals, whereas CS exhibits continuous variable changes throughout the simulation (Özgün & Barlas, 2009).

2.2.1 Discrete Event Simulation (DES)

Discrete-event simulation (DES) is defined by alterations in the simulation's state at specific, distinct time intervals. These intervals are called events, and they are laid on the time steps where it is assumed that the system remains unchanged between the events. In other words, DES comprises states and events interacting in a system. This type of simulation is well-suited for studying complex systems where events occur at specific, identifiable points in time, such as manufacturing processes, transportation systems, or computer networks.

To develop a DES, the requirements of the simulation should be identified in the beginning leading to a quantitative problem. The problem should be formulated as the next step. At this point, the model is formed based on the selected fidelity. The boundaries are set, and study objectives and specific issues are defined. Performance metrics for comparing model configurations are established, and key configurations of interest are outlined. The study's timeframe is decided upon.

Afterward, the real system data should be gathered to support the inputs and provide the existing performance. In case of uncertainty involvement, the probability distributions should be assigned to the inputs based on the data. At the same time, an appropriate software package should be opted to implement the simulation considering the problem area (Maria, 1997). There are numerous commercially available packages for different areas such as military, healthcare, supply chain

management, and so on. As the next step, the problem should be demonstrated as the processes by meaningful events. These events are represented by process blocks. Figure 2.1 shows an example of process blocks in a DES implementation for a bank service problem. In this setup, the customer, serving as the simulation entity, navigates through various processes within the system, interacting with different components and triggering specific events.

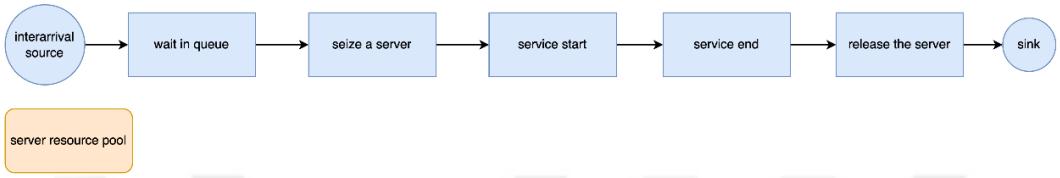


Figure 2.1. A sample of DES implementation

DES is also widely used in construction to detect project bottlenecks and ameliorate operations. By presenting CYCLONE, a network-based system, the simulation has been used as a tool to address the management complexities and identify the emergent outcomes in construction projects. CYCLONE was initially utilized to assist the planners with the scheduling process (S. AbouRizk et al., 2011). Later, other extensions such as UM-CYCLONE (Ioannou, 1990), Micro-CYCLONE (Lutz et al., 1994), and other systems like STROBOSCOPE (Martinez & Ioannou, 1997) and Simphony (S. AbouRizk & Mohamed, 2002) were developed to improve the management of construction projects (Figure 2.2).

In DES, a top-down modeling strategy is employed, envisioning a centralized authority that controls the various components within the simulation run. These components operate within predefined parameters and are unchanged within the events leading to a lack of ability to adapt or respond to changes in their environment. They function based on the instructions given to them and do not possess the capability to make independent decisions or influence the overall system outcome. This structured approach is beneficial for modeling systems with well-defined processes and clear hierarchical control but may not capture the complexities and

emergent behaviors present in more dynamic or decentralized systems such as construction sites.

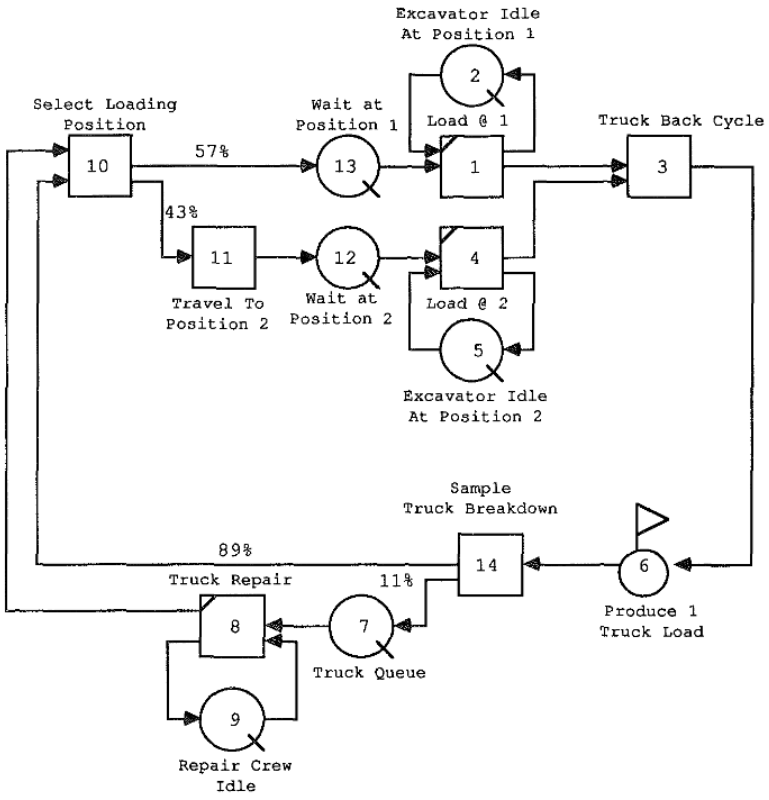


Figure 2.2. A CYCLONE model for earthworks (S. M. AbouRizk & Halpin, 1990)

2.2.2 Agent-Based Simulation (ABS)

Agent-based simulation (ABS) or multi-agent simulation (MAS) is a newer approach that's gaining popularity for tackling complex systems made up of independent, interacting agents. These agents operate on the determined rules, each with its own set of behaviors. They interact with the environment and each other, learn from interactions, and can adjust their behaviors to better fit their surroundings over time. By focusing on individual agents in the model, we get a clearer picture of the diverse attributes and behaviors that contribute to the system's overall dynamic behavior with a bottom-top approach (Macal, 2010). This approach better fits the systems consisting of uniquely behaving entities that can have a considerable impact on the

overall system performance such as construction sites (Sawhney et al., 2003; P. Zhang et al., 2019). Depending on the system type, the agents can represent countries, companies, crews, products, vehicles, and so on. They can interact with others, perceive the environment and other agents, and behave intellectually according to the assigned attributes. ABS's capability to distinguish each unique agent allows for the creation of a vast array of scenarios. While this can demand significant computational power, it also increases the likelihood of identifying more efficient solutions compared to other simulation methods (Khodabandelu & Park, 2021).

An agent possesses a state that changes over time. This state for an agent comprises key variables reflecting its current condition, which can be a full set or a selection of its attributes. The model's state encompasses all agents' states combined with the environmental state. An agent's attributes relate to its current state (Macal, 2010).

This simulation approach is also a good fit for visual simulations applied in entertaining computer games, but the main distinction is that in entertaining computer games there is no attempt to keep the rules based on the real system but rather unrealistic assumptions (Helbing & Balietti, 2012), however, ABS is essentially about building a digital version of a real-world system, depending on the detail level needed, to see how the system might respond when the agents interact. This has paved the way for a variety of computer-based visual tools, like serious games, training setups, and simulations for sectors like manufacturing and construction.

ABS can work seamlessly alongside other types of simulations creating hybrid simulations including discrete or continuous such as simulating evacuations during scenarios involving the spread of toxic gas. Similarly, this approach can be used to combine weather, environmental, or climate simulations with models depicting human responses to those specific conditions (Helbing & Balietti, 2012). In the construction sector, ABS finds applications across various stages, from pre-construction tasks like design, bidding, and planning, to actual construction activities

such as crew performance, and even post-construction phase like maintenance and emergency response management (Khodabandelu & Park, 2021). Moreover, by using ABS, the complex network of trade contractors and subcontractors that are indistinct from the project central authority can be simulated and analyzed. This advantage offers enhanced and dependable models and simulations in construction, boosting business operations (Sawhney et al., 2003).

2.2.3 System Dynamics (SD)

System Dynamics (SD) was introduced in the 1950s by Jay Forrester and is used to tackle large-scale, complex engineering challenges. It focuses on the interactions among different system components over time, capturing dynamics through elements like stocks, flows, and feedback loops. This approach offers insights into how systems evolve and behave dynamically over time (Tang & Vijay, 2001). More specifically, SD consists of the differential equations that are solved over time steps of Δt, using typically Euler method. A good example to understand SD is the SIR (Susceptible, Infected, and Recovered) epidemic model shown in Figure 2.3, Equation 1, Equation 2.

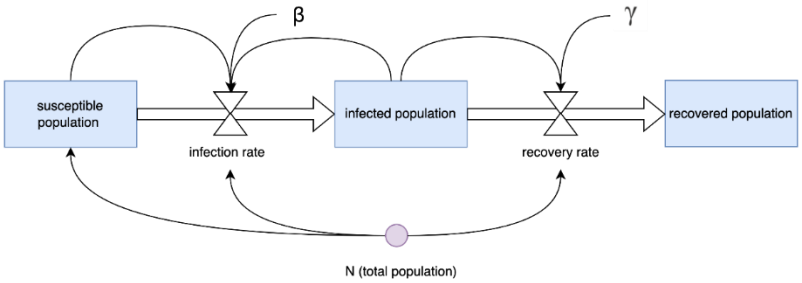


Figure 2.3. An SIR model in system dynamics

$$\frac{ds}{dt} = -\beta \frac{I}{N} S \qquad \text{Equation 1}$$

$$\frac{dI}{dt} = \beta \frac{I}{N} S - \gamma I \qquad \text{Equation 2}$$

β: infection factor

N: total population

S: susceptible population

This type of simulation is usually developed with deterministic variables and established around a network; however, it can be integrated with other simulation types. SD often involves creating models to understand complex dynamics and support learning and policy design, meaning that it is a good fit for macro simulation with large timeframes. SD proposes an approach called system thinking that provides a holistic view of a complex system to understand the connection of each part with the others to understand the actual system response against a decision (Sterman, 2002).

2.3 Anylogic as a Commercial Simulation Tool

Anylogic, as a commercially available simulation software, is capable of providing an object-oriented hybrid environment for DES, ABS, and SD simulation types (Figure 2.4). Hybrid models or multi-method models are used to mimic complex systems with constituent parts that demand different abstraction levels and modeling approaches. For instance, to support decision-making and model the interactions among the system constituent parts, ABS and SD multi-method can be preferred. A demographic model of a city including the behavior of city individuals regarding employment, housing, and relocations against the macro decisions by the authorities is an example of this approach.

With Anylogic, a modeler can select from a broad scale of abstraction levels, enabling efficient adjustments during model development (Borshchev, 2014). This flexibility allows for the seamless integration of various methods within a single model, providing a comprehensive and adaptable modeling environment. It uses Java programming language as the core of development with various ready-to-use process blocks (Figure 2.5). The model requirements and target fidelity determine the

method that should be used in Anylogic. Anylogic can be used in different areas such as supply chains, manufacturing, healthcare, logistics, construction, and so on. Moreover, it can be seamlessly integrated with external libraries supporting the development of the digital twin of the system.

Considering the simulation environment, Anylogic supports a limited extent of 3D model options. It uses ready-to-use 3D objects and also supports specific 3D file formats to be used as the environment. However, effective integration with the BIM model is lacking in this tool. In this case, whether the plan or sectional views should be used, which makes the simulation environment 2D, or an abstract 3D environment should be constructed using the available modeling tools in Anylogic.

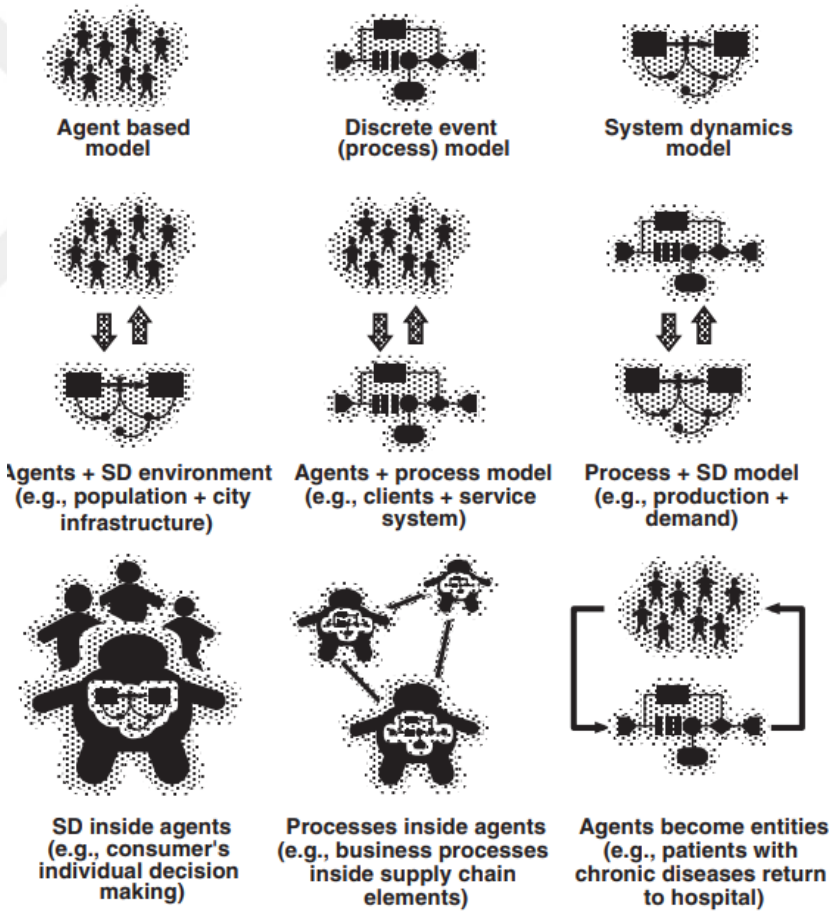


Figure 2.4. Hybrid simulation options using Anylogic (Borshchev, 2014)

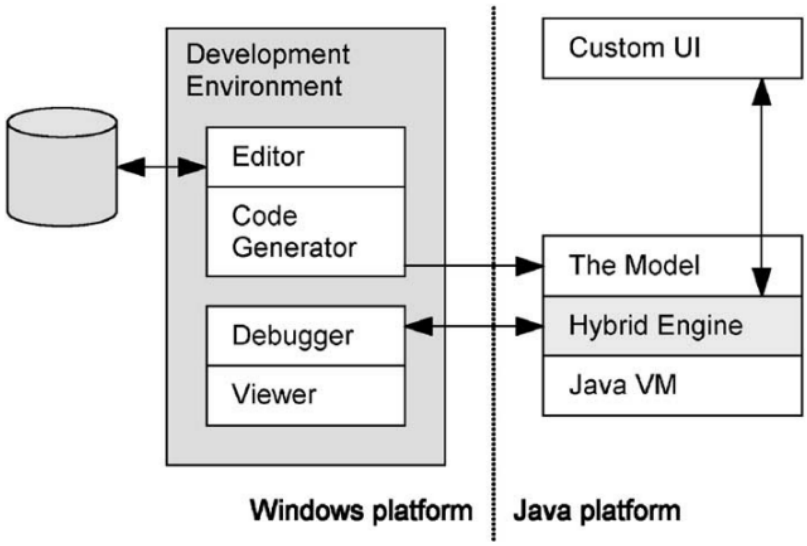


Figure 2.5. The architecture of Anylogic (Borshchev et al., 2002)

2.4 Game Engines

Despite the common entertainment-oriented use of game engines, they can be used to develop serious games, a simulation type with high fidelity, that is often valued when the goal is to apply knowledge gained in the game to real-world scenarios. The closer the resemblance between the virtual and real environments, the higher fidelity is achieved by the simulation. The game engine includes rendering, physics, AI, and game mechanics. It differs from graphics engines, which only handle rendering and Software Developer Kits (SDKs) that assist application development (Petridis et al., 2010).

Game engines, renowned for their ability to create high-fidelity 3D environments, are increasingly leveraged in educational settings. They serve as the backbone for simulators across diverse fields like flight training, medical procedures, construction workflow, and military exercises. These engines offer a cost-effective, risk-free virtual platform for conducting what-if scenarios and system simulations. Furthermore, game engines are instrumental in developing immersive virtual worlds that seamlessly integrate with virtual and augmented reality technologies (VR &

AR), enhancing user experiences by providing a heightened sense of realism and engagement (Mastli & Zhang, 2017).

The developed game engine simulation can integrate agents capable of distinct behaviors and attributes, empowering them to function independently within the virtual environment. These agents not only interact with one another but also adapt dynamically to evolving conditions within the simulation. This capability significantly amplifies the realism and intricacy of the simulated scenarios. Interestingly, this characteristic aligns closely with the fundamental principle of Agent-Based Simulation (ABS), where individual agents' interactions drive the system's overall behavior and outcomes.

2.5 Deterministic and Stochastic Approaches

With other perspectives, simulations can be classified as deterministic or stochastic depending on the degree of uncertainty involved. Deterministic simulations contain constant and stable conditions throughout the experiment. S. M. AbouRizk & Halpin (1990) confirmed that only a few construction operations can be considered deterministic while the rest should include some levels of uncertainty.

Hosny et al. (2020) investigated the challenges of factory design in manufacturing, shedding light on the shortcomings of current deterministic and mechanistic modeling approaches. These traditional methods often stumble capturing the dynamic and uncertain environments of construction sites. In response, the paper advocates a probabilistic emerging approach to facility modeling and control. This new approach recognizes the cumulative effects of uncertainty and the specific decision-making processes that construction teams exhibit in managing operational challenges. Through a comprehensive analysis that combines qualitative and quantitative analysis with industry evaluation, the research highlights the importance of these challenges and recommends a paradigm shift to more stochastic modeling approaches.

Feng et al. (2000) presented a new methodology for time-cost trade-off analysis of the construction industry by combining genetic algorithms (GAs) and simulation methods. This approach addresses the probabilistic nature of project networks and provides more realistic solutions under uncertainty. Research shows that GA, combined with simulation, provides an efficient and practical way to assess project time and cost risk, outperforming traditional heuristic and statistical planning methods. Probabilistic analysis helps project planners further support decision-making by allocating project duration and costs among different solutions over a period-cost trade-off curve, The paper concludes future development of time and cost trade-offs in construction projects under construction project uncertainty may wish to use a combination of GA and simulation.

Marschall et al. (2019) focused on the analysis of window openings, a specific aspect of occupant behavior. Research showed that natural ventilation has significant benefits for the sustainable design of the building and the comfort of its occupants. However, allowing residents to control their environment creates uncertainty about the effectiveness of the controls. Traditional energy simulation software models human behavior as deliberate, typically assuming windows are open at a set room temperature. Such simplicity creates a difference between simulated and real building performance. Accurate modeling of occupant behavior can help architects design and engage sustainable spaces and improve the accuracy of physical design. Recent field studies have examined adaptive control behavior to provide robustness examples have improved. They aimed to incorporate a data-driven stochastic model of window function into the original architectural designs. This approach helps designers understand the impact of occupant behavior on building performance and interpret uncertain outcomes from probabilistic forecasts.

2.6 Workspace in Construction

Space on construction sites is an important resource, which is essential for activities. Failure to manage departmental space demands can lead to time-space conflicts,

where the space demands of one activity interfere with the needs of another (Akinci, et al., 2002). Typically, space as a resource is not effectively considered in traditional scheduling methods such as the Critical Path Method (CPM) (Chavada et al., 2012). CPM is a mathematical analysis technique in which the early and late dates for the start and finish of activities are established based on a sequential network. In this technique the duration of the activities is deterministic and the primary aim is to compute float to identify activities with minimal scheduling flexibility (Project Management Institute, 2003). The activities with zero float are considered to be on a critical path, meaning that they drive to the project's finish date and can lead to delays. Space constraints are among the primary factors contributing to delays in critical path activities due to a decline in activity-level productivity (Guo, 2002).

2.6.1 Classification of Workspaces

Construction-demanded workspace for the crews can be classified into two main categories static and dynamic spaces. Static workspaces are the fixed spaces during the involvement of the crew in the assigned tasks. Storage of materials, fixed tool equipment space, and safety space are considered static workspaces. On the other hand, working space, loading & unloading spaces, and traffic space are dynamic workspaces. Additional classifications are applied based on previous studies within these overarching categories. These classifications aim to add more fidelity to the developed simulation by defining patterns specific to each class. Table 2.1 indicates the classifications suggested by a few previous studies.

Table 2.1 Classification of workspaces according to the previous studies (Messi et al., 2022)

Study	Workspace classifications
(Thabet & Beliveau, 1994)	Objects | Manpower | Material | Work blocks
(Akinci, Fischer, et al., 2002)	Component space | Labor space | Equipment space | Temporary structure space | Protected space | Hazard space
(Dawood & Mallasi, 2006)	Product space | Workspace | Path space | Storage space | Process space | Protected space | Support space
(Chua et al., 2010)	Process space | Interdiction space | Dead space | Resource handling space | Product space
(Moon et al., 2014)	Installation space | Transfer space | Loading space | Prefabrication space | Safety space
(Kassem et al., 2015)	Object space | Main space | Support space | Safety space
(S. Zhang et al., 2015)	Building component space (BCS) | Worker space | Material handling path | Protective space | Equipment
(Mirzaei et al., 2018)	Product space | Labor crew space | Equipment space | Temporary structure | Material storage space | Hazard space
(Getuli et al., 2020)	Worker space | Equipment space | Safety space| Hazard space

2.7 Influence of Time-Space Conflicts on Productivity

Productivity is a multifaceted concept encompassing production rate, performance, effectiveness, and the output of units within a given time frame. Traditionally, it is understood as either the ratio of input to output or conversely, the output relative to one or more inputs within a system (Dozzi & AbouRizk, 1993; Huang et al., 2009). For example, in the context of a construction project, labor productivity may be quantified as the manhours expended per tonnage of steel erected, or conversely, as the tonnage erected per unit of labor time spent. Meanwhile, for equipment and machinery, productivity can be gauged through the "effectiveness" factor, denoting the proportion of working time to the combined total of working and non-working

durations (Dozzi & AbouRizk, 1993). In construction projects, the complex nature of tasks, interactions of crews, and the unique dynamics of workflow conditions render traditional productivity measurement methods prone to inaccuracies. For instance, in finishing tasks, such as plastering, where one crew hands over completed work to another, even minor deviations in finishing time or quality can ripple through subsequent tasks, impacting overall productivity at the project level. In another case, material delivery machinery, like cranes or conveyors, wield significant influence over the efficiency of other crews, particularly in large-scale projects (Tantisevi & Akinci, 2008).

In contemporary business landscapes, productivity stands out as a principal factor of success. Given the constraints of time, budget, and natural resources, maximizing productivity is indispensable for maintaining competitiveness across all sectors, including the construction industry, which holds a significant share as a driver of economic growth in Europe (Nazarko & Chodakowska, 2015). Within academia, there has been an interest in exploring avenues to detect the origin of productivity declination and enhance it by examining the contributing factors and quantifying their impact. A fundamental prerequisite for productivity improvement lies in the ability to accurately measure it, thereby enhancing the reliability of estimations (Dozzi & AbouRizk, 1993).

Akinci et al. (1998) considered time-space conflicts as one of the main sources of productivity reduction in construction projects and introduced a framework that proactively manages spatial conflicts in construction by identifying and analyzing time-space conflicts before they occur. Sanders (1998) reported that conflicts can diminish the daily productivity of the crews by up to two-thirds of the estimated value and highlights the importance of quantified conflicts in construction projects. In this study, thirteen categories of conflicts were introduced and an analysis of variance with adjusted daily productivity was performed to quantify the impact of conflicts on productivity. The analysis underscores the substantial impact of congestion on both project productivity and schedule. The importance of meticulous planning and effective control measures to mitigate the occurrence and mitigate the

adverse effects of disruptions is emphasized. Kuntz (1994) described the operation space as one of the important factors in crew productivity and proposed a method called the Construction Crew Evaluation Model (CCEM) to improve productivity. Dabirian et al. (2023) proposed a simulation method to predict the productivity change of the crews under space conflicts. The approach employs an Agent-Based Simulation (ABS) method, utilizing an environment structured around the Moore Neighborhood model with a range of 3. The agents within a range of 3 or less are considered to have conflicts that negatively impact productivity. They defined productivity as the overall performance measure of a workforce on-site considering their skill level. Meanwhile, their productivity was determined by comparing the actual work performed to the expected output based on their skill level. The developed simulation reported a 30% reduction in productivity due to space conflicts.

Khataei et al. (2020) presented a methodology that uses 4D BIM and a visual programming tool to modify the value results of construction activities by calculating the proximity of the objects. The method generates revised productivity value and rescheduled plans, resulting in more reliable baseline estimates for the sake of increased estimation reliability. Unlike traditional methods that assume constant productivity of resources, this method accounts for variable productivity values depending on the physical proximity of the objects. Therefore, it was aimed to improve the accuracy of long-term analysis as well as preliminary estimates.

2.8 Use of Simulations in Construction

2.8.1 Process-based Simulations

The introduction of CYCLONE by Halpin (1973), a network-based simulation system, has been one of the first uses of simulation in solving complex construction problems. Initially implemented to aid planners in scheduling (S. AbouRizk et al., 2011), its utilization has spurred the development of subsequent extensions like UM-

CYCLONE (Ioannou, 1990), Micro-CYCLONE (Lutz et al., 1994), alongside other systems such as STROBOSCOPE (Martinez & Ioannou, 1997) and Simphony (S. AbouRizk & Mohamed, 2002). These platforms, primarily grounded in DES type, are engineered to improve project operations by intricate modeling system processes, thus advancing project management within the construction domain.

Lu (2003) presented a new method called the Simplified Discrete Event Simulation Approach (SDESA) developed for architectural simulation, which aims to match the weaknesses of the Critical Path Method (CPM). It explores the basic principles of discrete event simulation and its algorithm and introduces SDESA by reviewing the model and program. The effectiveness and simplicity of SDESA are demonstrated through two real-world case studies of road construction in Hong Kong. These findings highlight the ability of SDESA to model complex systems effectively. In addition, SDESA is designed as a versatile production planning tool, adept at solving challenges beyond CPM-based network analysis. It also provides specialized simulation tools for handling large and complex instrumentation systems.

Kim & Kim (2010) developed a multi-agent simulation system to replace the conventional discrete event simulations to appraise the traffic flow among the construction equipment. In the mentioned case, the vehicles are required to set their attributes based on the perceptions of the other vehicles or the environment that is not supported by the DES simulations. They demonstrated the capabilities of multi-agent simulation in the modeling of the problem and detecting the negative effect of congestion on the efficiency of the equipment.

Alvanchi et al. (2011) proposed a hybrid simulation method that integrates SD with DES for manufacturing systems. The simulation aims to capture the dynamic behaviors of manufacturers and their interaction effects by considering both operational and contextual variables. The proposed hybrid method and associated architecture address challenges such as the lack of a modeling framework, advanced time-related issues, and communication scheduling. The effectiveness of the model is tested through a case study in a real steel fabrication shop, which shows that the

model is able to capture the actual process with improved simulation run times, communication strategies, and ease of use. The research shows that this hybrid model can be used to create a virtual fabrication enterprise to enhance the design analysis capabilities of construction projects.

Ding et al. (2018) presented a comprehensive review of system optimization (SD) and agent-based modeling (ABM), two key approaches to addressing complex construction waste management (CWM) systems regarding the advantages and disadvantages varieties of these two. After consideration, the paper proposes a new framework that combines their strengths to better tap into the subtle strengths of CWM. By presenting two integration options for SD and ABM, the aim is to enhance CWM system theory by incorporating macro-level dynamics and micro-level interactions. They highlight the urgent need for better management of construction waste and explore the potential of a hybrid approach combining SD and ABM.

Wang et al. (2019) proposed an exemplary agent-based analysis of safety management behaviors in small enterprises (SSEs), focusing on the impact of employee safety management behaviors (EPOS-PB) such as whistleblowing and public sleepwalking. It highlights the important role of these control behaviors in improving safety outcomes through interactions between employees, citizens, and government. The survey uses an employee questionnaire to reinforce the model and it suggests that promoting EPOS-PB can lead to better protective support from the public and government.

Kedir et al. (2022) presented a new approach, using agent-based modeling (ABM) paradigms combined with reinforcement learning (RL). This methodology is used to simulate the architectural design process and then optimize it within the constraints of the project. The proposed framework facilitates informed decision-making by offering an optimized sequence of work packages, taking into account the complexities of project resource planning. The operationalization of this approach involves the transformation of construction planning into a network graph, upon which both ABM and RL algorithms are applied to ascertain the optimal solution.

This integrated approach contributes to more efficient and effective management of the construction project works by using ABM and RL in interactive graph systems.

Attajer & Mecheri (2024) developed an ABS method to improve the supply chain of modular integrated construction elements (MiC) to enhance efficiency and environmental effects. Their method uses agents involved in transport logistics to detect emergent behavior of the system against various scenarios aiming to minimize the transport cost and avoid delay penalties. The method is implemented in Anylogic with a hybrid simulation approach (Figure 2.6). Their findings highlighted the contribution of ABS to the improvement of the MiC supply chain.

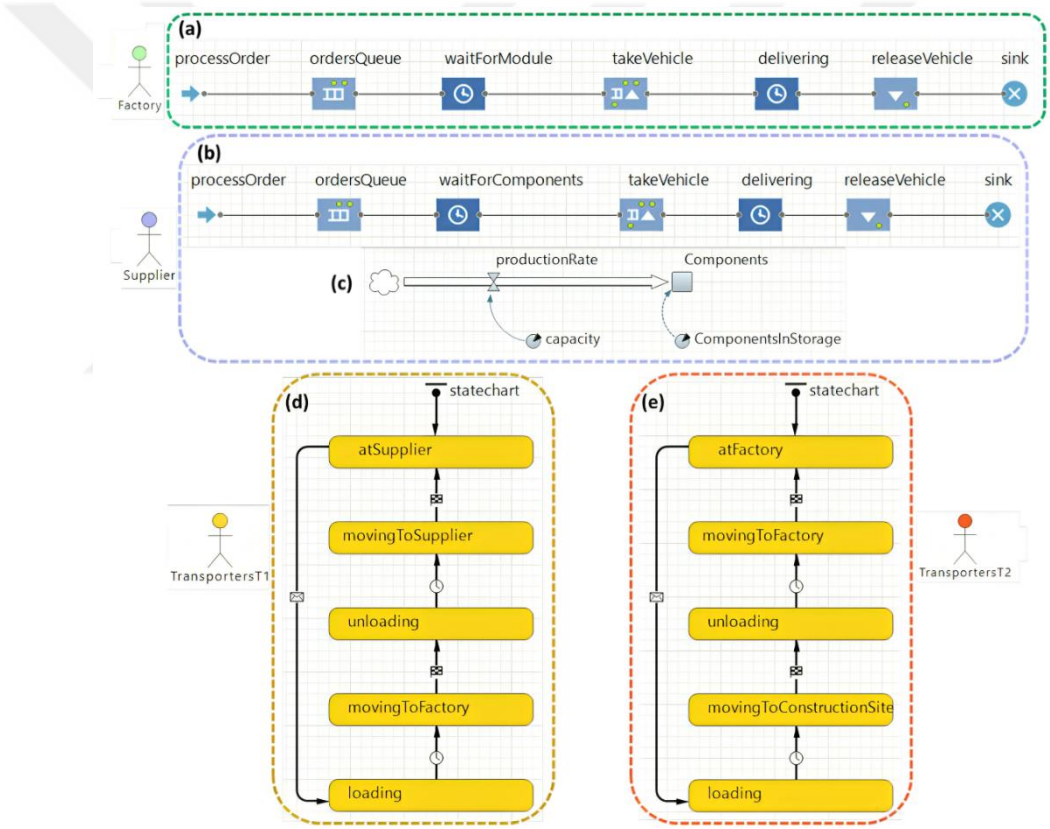


Figure 2.6. The simulation environment to model MiC transport logistics (Attajer & Mecheri, 2024)

Luo (2024) proposed an SD model to appraise the effect of the macro decisions on prefabricated buildings (PB) (Figure 2.7). The importance of capital investment, tax

incentives, and governmental demand are highlighted to manage the negative effect of construction on the environment and national economy. They focused on China as a case project to optimize the policies in PB growth.

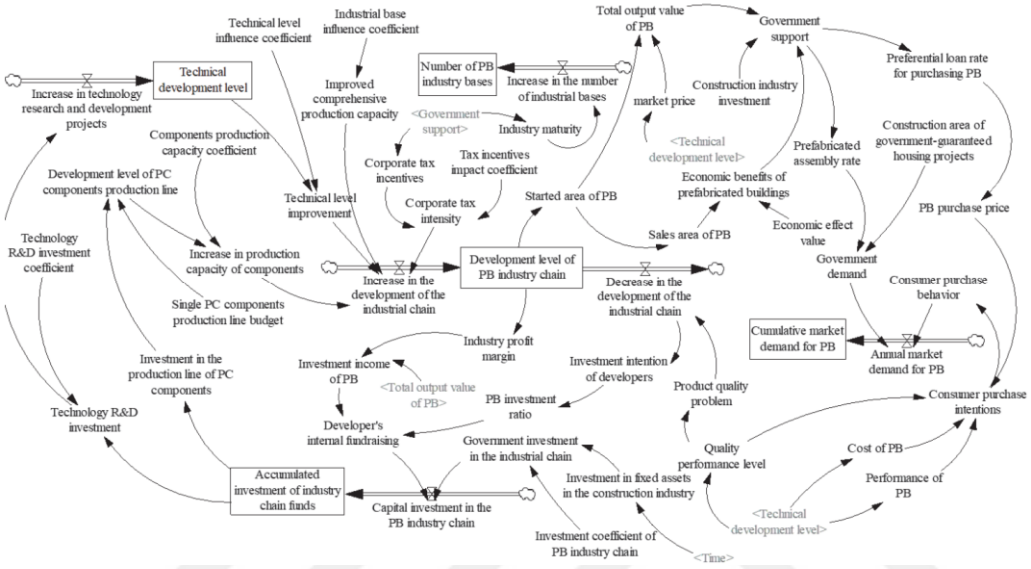


Figure 2.7. The SD model developed for the optimization of policies for PB (Luo, 2024)

2.8.2 Visual-based Simulations

Visual simulation leverages cutting-edge computer graphics techniques to enhance the creation and implementation of simulation experiments. With advancements in computer technology, graphic-based approaches for simulation have become a common method for modeling real-world systems alongside traditional programming. Modelers can create models visually rather than writing simulation code in programming languages. This graphical representation makes it easier for users to understand the models, improving communication and comprehension compared to models presented as computer programs (Hong et al., 2002). Visualization is increasingly popular in construction because it enhances understanding of the construction process and performance. For effective decision-

making, these visual models should be connected to project information (Moghadam et al., 2012).

2.8.2.1 Building Information Modeling

Building Information Modeling (BIM) is a digital model of the project with an object-oriented approach that involves the precise geometry of the model elements with additional data required for fabrication, construction, operation, and even demolishment. BIM was developed on the concept of the Building Description System (BDS) proposed by Charles Eastman in a study published in 1974 (Figure 2.8). In this work, a library of parametric objects is developed to model the building structures. In BDS, the library acts as a database alongside the analyzing unit to help designers achieve a holistic model with parametric behavior (Eastman et al., 1974).

Later, BIM found its way through the design process by enhancing interoperability in the modeling stages, providing the freedom of using different software packages for various purposes in the same project. Interoperability stands for the ability to share the model across different platforms without depending on the specific software package.

The mindset behind BIM is to develop an object-oriented model in which the objects are the instances of the classes with associated fields. For instance, an interior partitioning wall is an instance of the wall class with fields such as material, thickness, and inner structure to distinguish it from other wall objects. This hierarchy of objects can also be found in interoperability model formats such as Industrial Foundation Class (IFC) (Figure 2.9). In this context, the data model obtained from BIM enables the integration of the model with the various simulations including the ABS simulation type where each model object exists independently with distinct properties and interacts with the model agents.

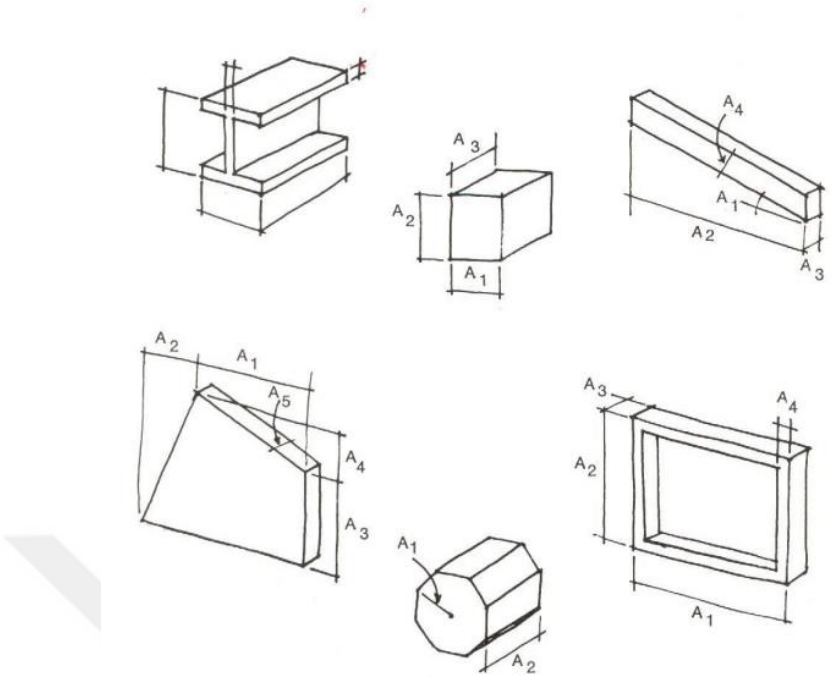


Figure 2.8. A first overview of BDM elements suggested by Charles Eastman (Eastman, 1975)

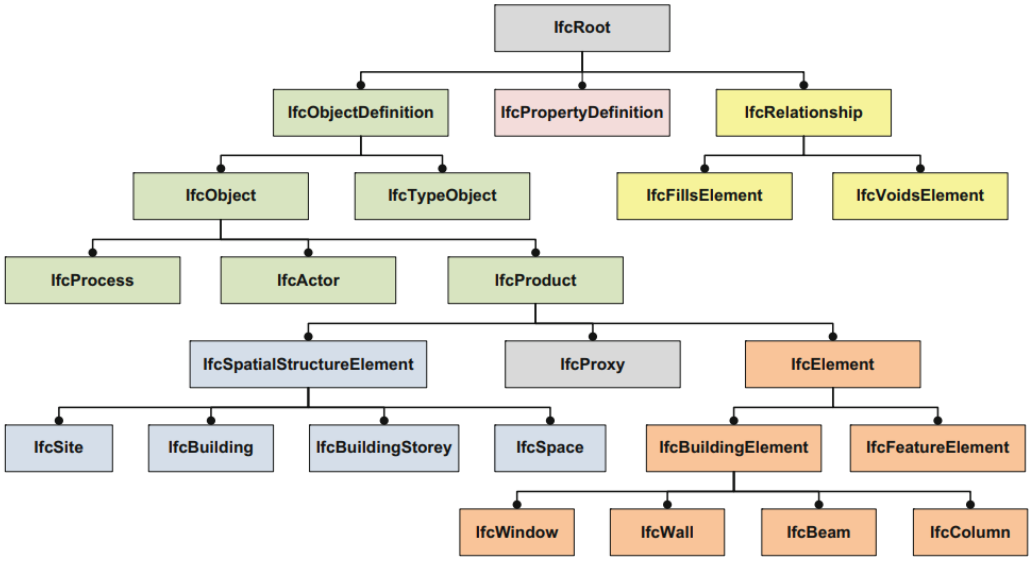


Figure 2.9. An example schema of IFC (Mendes Da Luz Patacas et al., 2014)

Zaman et al. (2024) proposed a method to integrate BIM with a gaming engine to enhance construction safety management through two case projects. The first project concerns the scaffolding failure scenario within a construction stage and the second concentrates on the evacuation plan in the pre-construction stage. The aim of the study is to enhance the safety of the projects by simulating the scenarios in the interactive gaming environment.

Au (2024) investigated the advantages of integrating game engines with architectural design iterations and client involvement. The work addresses the gap between the project state and the client's knowledge by improving the demonstration of design in an interactive and immersive simulation environment (Figure 2.10). The generated methodology implies the effective potential for improvement in architectural design experience.

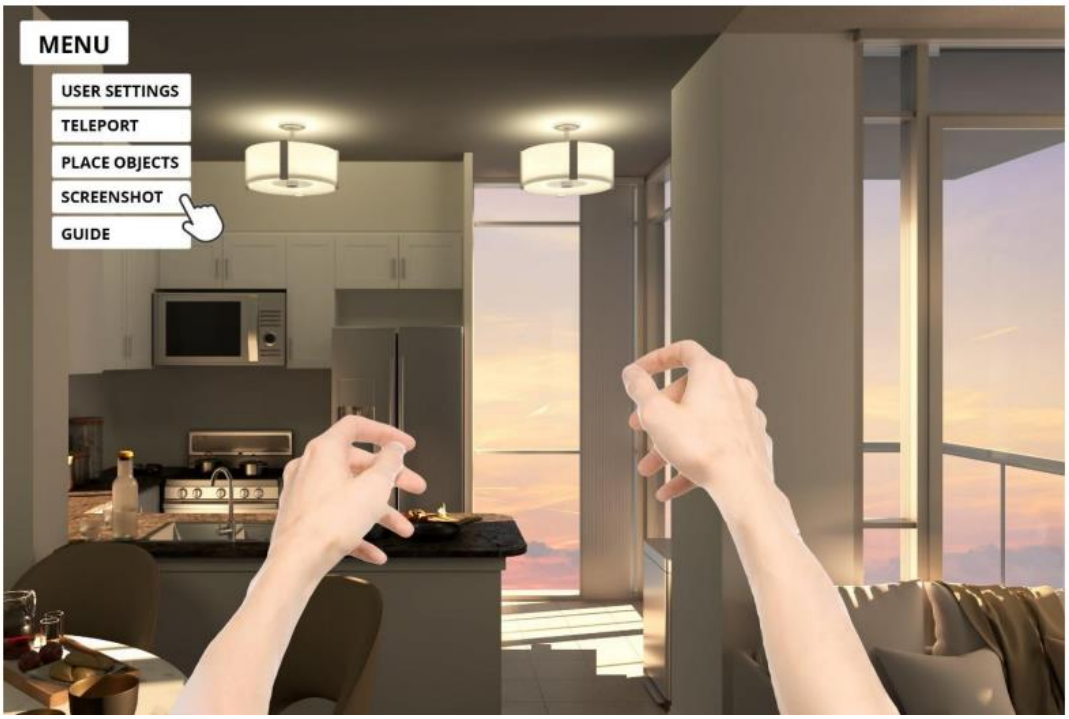


Figure 2.10. The integration of the gaming engine with architectural design (Au, 2024)

2.8.2.2 Visual Programming Language

Visual Programming Languages (VPL) are a type of programming manner where the functions and operations are represented by ready-to-use script blocks. They ease the way a program is developed by minimizing the tedious coding from scratch processes and help the user better understand the structure of the associated framework and available syntax within a project.

In the BIM environment, VPLs are used to handle tedious repetitive work, generate complex graphs, query model data, and perform several analyses by integrating the model with other environments. Considering the user interface (UI), VPLs typically consist of two sections: a working space and a library of nodes. The nodes are the pre-built functions or operations that can connect using the links. The start and end of the links signify the connection of the output of a node to the input of the subsequent node (Figure 2.11).

The level of granularity is a distinctive factor among the VPLs that describes the level of access to each constituent part of the node codes. In other words, the level of granularity signifies the encapsulation level of the nodes and determines whether the sub-scripts in a node are available to provide more control to the user for advanced coding (Mendes Da Luz Patacas et al., 2014). The higher encapsulation limits the access of the user to the sub-scripts yet simplifies the programming environment. Therefore, for deeper access to the BIM data model requirements, VPLs may not be an effective solution in this context.

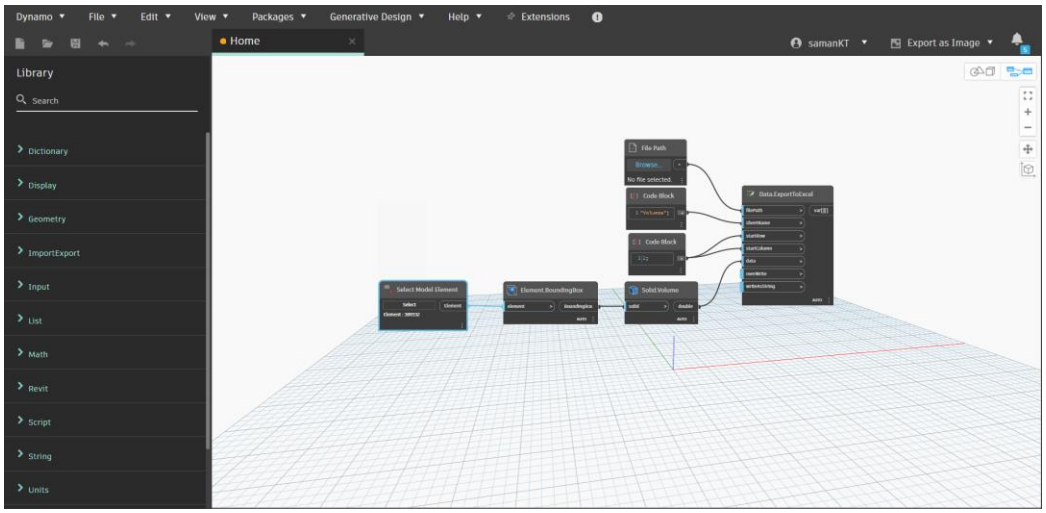


Figure 2.11. A VLP user interface with the node library (left) and workspace (right)

Table 2.2 shows the list of the selected list of the VLPs and their use cases. Autodesk Dynamo is a well-known open-source VLP seamlessly integrated with the Autodesk Revit software package. It has access to the model data using its specific syntax and Iron Python framework. In the current study, Dynamo is used to perform queries regarding the model data and crud operations on the external databases. Moreover, it is used to create new parameters on the model and assign values to them to be used during the simulation runs. For instance, the element IDs are appended to the name of the model elements aiming to better map the model objects with the database entities in the simulation environment.

Table 2.2 A list of the selected VLPs and their use cases

VLP title	Manufacturer	Use case
Dynamo	Autodesk	Autodesk Revit, standalone
Snap	UC Berkley and SAP	Web application
AgentCubes	Alexander Repenning	Web application
StarLogo	MIT	Java

Along with a sophisticated 3D model provided by building information modeling (BIM), the data notion added to the model is the most valuable aspect of this

technology. This combination adds layers of information that help in planning, executing, and managing construction projects. To be more specific, BIM is a modeling approach in an object-oriented mindset that is packed with useful data, making everything from design reviews to facility management a whole lot smoother and more precise. BIM alongside a Common Data Environment (CDE) acts as a core that brings together various disciplines to be involved in the project lifecycle and collaborate more effectively. By providing a unified platform where architects, engineers, contractors, and other stakeholders can share and access information in real time, BIM fosters better communication, coordination, and integration across all stages of a project. This collaborative approach ensures that everyone is aligned with the project goals, leading to more cohesive designs, efficient workflows, and successful project outcomes.

Furthermore, by integrating the time notion, the generated BIM model evolves into the 4D BIM model, which hosts scheduling data alongside the corresponding model objects. This dynamic model serves as a valuable tool for tracking project progress and conducting time-sensitive analyses, empowering stakeholders to grasp constructability intricacies and anticipate potential risks more effectively. Moreover, it is used to develop a dynamic simulation called 4D BIM-based simulation incorporating advanced features for enhanced functionality in construction projects. In this simulation type, the progress of the project is demonstrated by linking the model objects to the time points. So, at each point of project time, the associated model object is displayed on the model. This way, the user can examine the possible conflicts and constructability issues (Figure 2.12).

Hong et al. (2002) introduce a user-friendly visual environment for architectural design modeling, which simplifies the process by removing the need for programming. It highlights the Activity-Based Construction (ABC) modeling method, which employs a single type of element, the activity. This makes the graphical model more like a Critical Path Method (CPM) diagram, making it easier to understand. In addition, the paper includes a straightforward animation method that uses 2-D reputed images of a graphical simulation model as a background, which

supports model validation. The system provides dynamic calculations and visual representations of simulation results of manufacturing operations being modeled internally to enhance user understanding.

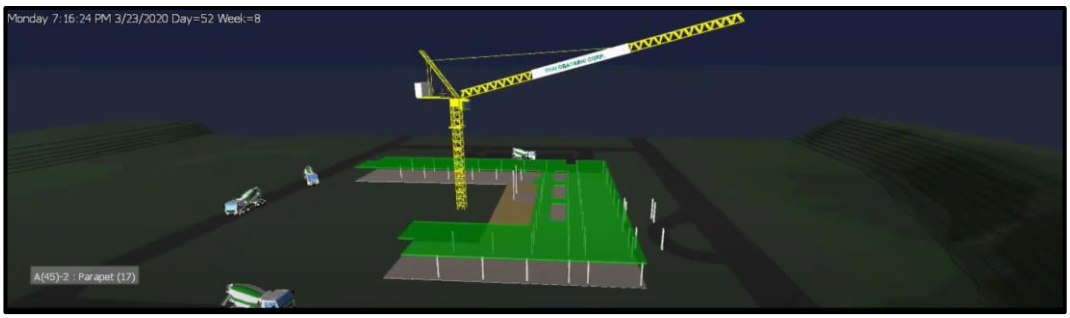


Figure 2.12. A 4D BIM-based simulation scene

Arturo Osorio-Sandoval et al. (2020) developed a BIM-based DES implemented on a game engine to enhance visualization and usability in construction projects. Their system includes four modules: environment, simulation, user input, and animation modules. The animation module is responsible for rendering the model object based on the BIM model aiming to improve verification and planning of construction activities. Mixed reality is also supported for further visual analysis.

Akpan et al. (2020) investigated the impact of 2D display and 3D visualization/virtual reality (3D/VR) on the success of simulation projects. It considers 193 Success Statements (SSs), 5 Critical Factors (CSFs), and 15 Key Performance Indicators (KPIs) to assess project outcomes by the mentioned simulation approaches. Using text analysis and sensitivity analysis, the study uses sensitivity from literature in line with SS, CSF, and KPIs. The findings show that 3D/VR has a greater impact than 2D displays on both KPIs and CSFs, contributing significantly to the success of simulation projects.

Pérez et al. (2016) offered a comprehensive analysis of the way 4D Building Information Modeling (BIM) enhances logistics operations and facility control inside the production enterprise. It highlights the benefits of 4D BIM in terms of simple construction principles that help by reducing non-value-added activities and

improving the spatial design process. The authors identify notable differences in design logic and the use of 4D BIM for logistics and identify areas for future research. The aim of the paper is to offer a comprehensive knowledge of the modern state of 4D BIM in construction design and facility control.

Kamat & Martinez (2001) highlighted the worth of 3D visualized simulations in improving project design, especially for complex construction tasks and decision-making. They introduced a simulation system called Dynamic Construction Visualizer (DCV) to provide a comprehensive 3D visualization methodology for simulation models in construction, enhancing the verification, validation, and reliability of the models. This system converts discrete events from simulation models into dynamic continuous motion, allowing users to visualize complex construction operations. The DCV is designed to be independent of any simulation tools and uses a text file-driven approach to generate 3D animations, which are processed from sequential, time-ordered commands written in a simple ASCII-based language.

To visualize construction operations, DCV imports 3D models from external CAD sources and supports programs such as VRML to ensure high-quality and versatile modeling. This approach avoids the limitations of built-in CAD capabilities and allows the use of commercial-grade prototypes. By providing a realistic and accurate 3D model of production design, DCV greatly improves the utility of simulation as an analysis tool in construction planning and operations (Figure 2.13).



Figure 2.13. A snapshot of the developed simulation (Kamat & Martinez, 2001)

Al-Hussein et al. (2006) developed a system that integrates 3D modeling and special-purpose simulation (SPS) for tower crane operations on construction sites, to improve planning and decision-making processes. The system addresses the challenges of using simulation in construction processes, such as the operational complexity and tools needed to enable construction managers to use it effectively. The benefits of 3D visualization are emphasized, particularly its role in verifying and validating simulation results and effectively communicating complex operations among stakeholders. A case study involving the construction of the University of Alberta's new engineering building demonstrates the practical advantages of combining 3D visualization with SPS for tower crane scheduling (Figure 2.14).

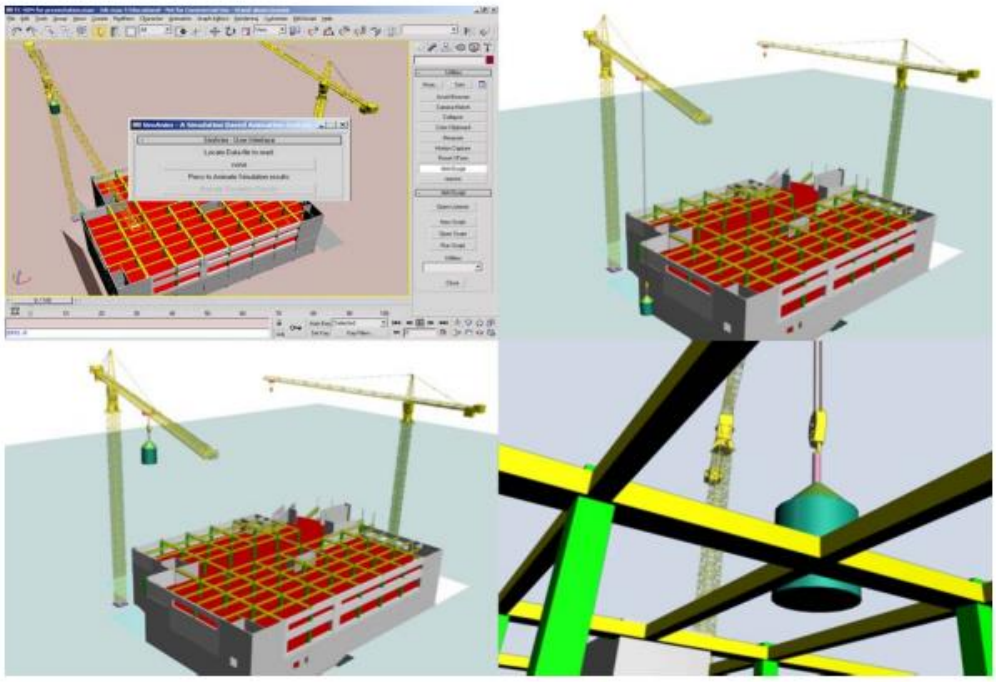


Figure 2.14. A scene of the generated simulation (Al-Hussein et al., 2006)

Moghadam et al. (2012) explored the advantages of factory-based modular construction, emphasizing how this method enhances pace, quality, and safety while handing over widespread value and a quick return on investment. The method underscores the vital role of simulation inside the early stages of a project, providing a methodology that mixes simulation with post-simulation visualization to aid in scheduling the onsite installation of prefabricated modules.

A notable case study involves a 34-story building in Brooklyn, New York, where the methodology was applied. The simulation model output was transformed into a visual animation to effectively communicate the schedule and identify efficient solutions. The study highlights the synchronization of construction activities, which enabled the onsite installation to be completed in just 56 working days. The paper concludes that the proposed post-simulation visualization (PSV) model is highly effective in planning and executing modular building construction, leading to improved schedules, productivity, and decision-making during the planning phase.

2.9 Project Delay Management

The complex nature of construction projects and unpredictable incidents during the project execution make it inevitable for schedules to deviate from initial estimations, causing frequent delays in the projects. Some of the main reasons for delays are financial issues, material shortages, change orders, weather conditions, and improper project management (Indhu & Ajai, 2014). In these circumstances, a Root Cause Analysis (RCA) should be conducted to identify the delay's underlying factors comprehensively (Project Management Institute, 2003). This step is followed by the appropriate alleviative actions through the revised schedule to expedite the project and avoid the delay costs.

2.9.1 Project Acceleration Methods

In this context, two commonly utilized methods for schedule compression are fast-tracking and crashing. Fast-tracking is a method used to speed up schedules by having activities or phases that can run concurrently, at least for some time. For instance, this could mean starting the construction of a building's foundation before finalizing all architectural drawings. While fast-tracking can reduce project duration, it may lead to rework and heightened risk. It's effective only when activities can be overlapped without delaying the critical path. However, implementing fast-tracking often requires increased coordination efforts among involved activities, which can elevate the risk of compromising quality. Crashing is a strategy employed to reduce the overall schedule duration with minimal additional cost by injecting extra resources. This could involve actions such as approving overtime, sustaining the workforce, or paying to accelerate the delivery of materials or services for activities crucial to the project's timeline. The effectiveness of crashing lies in its focus on critical path activities, where injecting more resources can downsize their duration. However, it's important to recognize that crashing isn't always a feasible option and might introduce heightened risks or expenses (Project Management Institute, 2003).

As allocating more resources through crashing entails extra expenses, it becomes necessary to expedite critical path activities to minimize overall project duration. In real-world projects, decision-makers consistently engage in a delicate balance between project time and cost through an analysis called the Time-Cost Trade-Off Problem (TCTP) by considering the alternatives produced by crashing to reduce the duration of activities. Researchers widely interpret the TCTP as a critical decision-making model in project management, emphasizing its significant influence on project success (Biswas et al., 2016).

As the project resources are generally in discrete units, the model is considered a discrete time-cost trade-off problem (DTCTP), where the model variables are the crashing time-cost alternatives called modes (Figure 2.15). In practice, project planners often rely on their experience or past project data to opt for DTCTO alternatives. However, construction projects, each with their unique characteristics, can produce different results even with similar resource allocation. The availability of working space for project activities emerges as a crucial factor for these alternatives. It's important to recognize that as resources are increased, the dynamics among the project crews arise. Thus, a detailed analysis of space demand and availability becomes essential to obtain a reliable DTCTO solution.

2.10 Time-space Conflict Detection by Simulation Techniques

As discussed, although assigning more manpower and machinery resources to an activity is expected to shorten the activity duration, the reduction may not take place as expected and corresponding costs can overtake the predicted amount, mainly due to space conflicts among the crews resulting in productivity reduction (Said & El-

Rayes, 2013; Thomas et al., 2006). Forming crashing alternatives and finding optimal crew size, demands a rigorous appraisal of the project conditions to have realistic schedule and cost expectations for achieving project success, but typically, these alternatives are generated based on the project planner's insights or previous project records. In this approach, mostly, the performance reduction caused by the workspace conflicts among the project crews or equipment is disregarded or misestimated which imposes a considerable deviation of the schedules from initial estimations (Hemalattha & Vidjeapriya, 2021; Tao et al., 2018) as well as can cause safety issues (Ballesteros-Pérez et al., 2019; Chavada et al., 2012; Pradhananga & Teizer, 2014; S. Zhang et al., 2015).

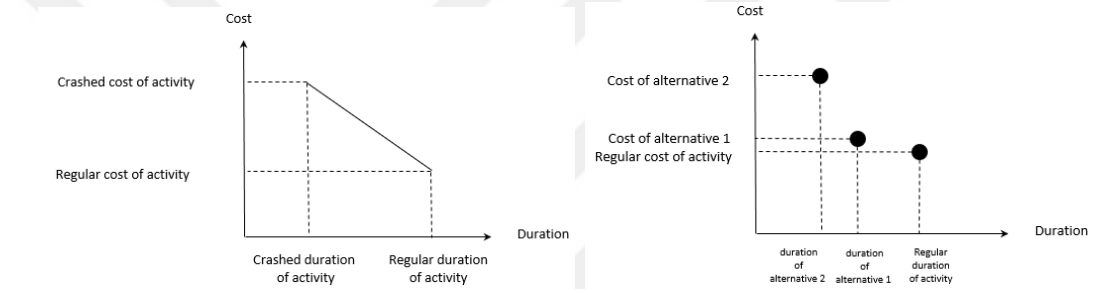


Figure 2.15. Examples of TCTP (left) vs DTCTP (right)

Since the number of crews sharing the same workspace grows, it's expected to anticipate an increase in conflicts, which ultimately lead to a decline in their performance (Mirzaei et al., 2018). Therefore, quantifying these space conflicts becomes crucial for devising more practical crashing alternatives (K.W. et al., 2012; Semenov et al., 2014). Moreover, given that conflicts evolve over time, it's essential to use a dynamic approach to space congestion analysis, which in turn forms time-space conflict problems.

Quantifying conflicts between time and space within the complex realm of construction projects poses a significant challenge due to the diverse array of factors. Firstly, each project possesses a unique set of circumstances, necessitating a tailored analysis that encompasses factors such as crew composition, construction

methodologies, and project layout. Secondly, the behavior of each crew within the spatial domain can vary, demanding precise modeling and assessment. Thirdly, time-space conflicts are inherently temporal in nature, reflecting the dynamic nature of crew activities, thus requiring an analysis framework that accommodates this dynamism. Lastly, given the inherent uncertainty within the construction environment, adopting a probabilistic approach is essential for conducting thorough analyses.

Conflicts among the crews within a project can significantly impede productivity, leading to deviations from the planned schedule. Hence, to enhance the accuracy of estimations, it is important to identify conflicts and measure their impact on productivity at the activity level. To date, numerous studies have tackled these complexities by presenting various methodologies for planning workspace and detecting conflicts in time and space. Thabet & Beliveau (1994) categorized the activities with required workspaces into three classes A, B, and C, representing activities that solely occupy the whole space, those that share space with other activities, and those that need space for material storage, respectively. A scheduling model was presented that takes the workspace into account for multistory buildings. In another study, an AI system called SCaRC – a Space Constrained and Resource Constrained knowledge-based system- was developed that considers workspaces and resources as two different constraints during the scheduling. In the system, workspaces are categorized into construction and material blocks, and an AI technique called knowledge-based expert systems was employed to handle the planning. Moreover, to include the efficiency loss during the space conflicts, a set of user-defined productivity-SCF (space capacity factor) curves was proposed (Thabet & Beliveau, 1997).

A system that detects time-space conflicts by employing CAD and schedule plans was developed by Guo (2002) in which the workspaces are categorized into working, storage, setup, and waste spaces. The study ameliorated previous research by incorporating path conflicts and conflict severity assessment into the analysis. The

detected conflicts were resolved by rearranging the space plans or revising the initial schedule.

Younes & Marzouk (2018) developed a 2D ABS to solve the tower crane problem considering time and cost aspects. Their proposed method was able to capture the interactions among the activities, appraise the effects of conflicts on the time and cost of the project, and optimize the location and number of the cranes. They showed that the generated method reduces the duration and cost of the project by minimizing congestion and providing decision scenarios to the project managers based on the unique conditions of the project.

Liu et al. (2022) developed a multi-agent system to simulate the construction actors to mitigate the time-space conflicts in the floor plane. The simulation is implemented in a 2D environment considering the agents' movement and their interaction. It aims to detect the spatial congestion among the agents in limited spaces of a project floor and analyze the corresponding effect on the project.

In the mentioned studies 3D models are not utilized for contributing spatial analysis, meaning that the space conflict analysis does not include model element postural data. Additionally, due to the unpredictable nature of conflicts in construction projects, where each one is complex and distinct, it's crucial to virtually simulate the system with sufficient accuracy to identify unforeseen conflicts before they arise (Dawood & Mallasi, 2006; Mallasi, 2006; Riley & Sanvido, 1997). The simulation should be dynamic and be capable of representing the real conditions of the project. 4D models in this respect have been widely applied to simulate the project environment so far.

Akinci, et al. (2002) addressed the basic problem of time and space conflict in the construction industry, which arises when the spatial requirements of one activity conflict with another, destructively affecting construction processes. A classification of time and space conflicts based on a 4D model called 4D WorkPlanner Time-Space Conflict Analyzer (4D TSConAn) is developed to identify and prioritize the conflicts in the initial stages of the project. This system brings together product and process

models with defined workspace requirements. The study builds on prior work in construction space management to provide a classification framework for time-space conflict analysis. It uses a space-loaded production model to automatically detect conflicts between four parameters (x, y, z, and time) and classify them based on the proposed classification. Furthermore, the paper presents a case study showing how to effectively implement the proposed system. This proactive approach can prevent delays and productivity losses in construction projects by addressing potential conflicts in advance.

Mawlana et al. (2015) proposed a method for planning and evaluating the phasing of elevated urban highway reconstruction projects. It combines 4D modeling and DES in a two-stage approach to create optimal, feasible phasing plans and assess potential space conflicts. This method addresses the challenges of complex geometry, limited space, and the necessity to maintain traffic flow and safety during construction. Key elements include using 4D modeling to merge 3D models with construction schedules for clash detection, employing DES to estimate durations with uncertainty, and a constraint-based system to ensure a clash-free plan. Additionally, stochastic clash analysis assesses risks under uncertain conditions and suggests optimal starting dates to minimize clashes.

The capability of the method is demonstrated through a case study of the Turcot Interchange reconstruction project in Montreal, Canada. The study shows how this approach can lead to conflict-free planning and the impact of uncertainty on the project schedule, providing valuable insights for traffic planners and project managers. The paper concludes that this approach has great potential to optimize project length and cost in terms of reducing social and environmental impacts.

Mallasi (2006) developed an approach for time-space conflict analysis using a 4D simulation as an add-in within a CAD tool. The simulation environment consisted of twelve execution patterns, three work-rate distributions, and a time-based progress simulation. In this way, the severity of the conflicts was evaluated by a multicriteria function, and the required workspace for each activity was generated by the

Approximation Envelope (AE) method, in which a bounding box over each activity is modeled representing required workspaces. This approach improved the previous studies by providing a dynamic representation of workspaces using the boxes adapting the position of the activities.

An experimental tool for simulating construction processes was proposed by Ben-Alon & Sacks (2017) that is based on ABS and BIM. The aim is to understand discipline-based crew behaviors by performing interviews and encapsulating them in simulation agents. The simulation models decision-making and situational awareness while using BIM to describe physical system conditions (Figure 2.16). Validation tests showed that the simulation tool could predict patterns similar to those on actual construction sites and explore complex scenarios. The method includes capturing a wide range of trade crew behaviors from field interviews and observations and expressing these behaviors as functions and rules in the simulation. The simulation was validated by simulating predictable scenarios and then applying it to more complex scenarios to test its feasibility. The study concludes with a discussion of the results, emphasizing the potential of the method for researching construction control methods in construction.



Figure 2.16. A simulation scene of the proposed simulation (Ben-Alon & Sacks, 2017)

A framework was generated by Choi et al. (2014) to identify the workspace conflicts and solve them by 4D BIM models and prearranged databases which are used to provide material properties and corresponding construction methods. The workspaces were classified based on the functional and movement categories where functional classification represents the purpose of the allocated workspace, such as object workspace, storage space, or pathing space, and classification by movement stands for the determination of whether a workspace is fixed, like a masonry material storage -which should be adjacent to the working place- or flexible such as plumbing equipment and material locations. In this manner, the project planner should devise a work execution plan to determine daily space occupation patterns for the crews to be used in the simulation.

In another study conducted by Mirzaei et al. (2018), a time-space conflict detection system was developed that determines not only the time-space conflict of the activities but also the severity of the conflict that can lead to a performance reduction or constructability problem. 4D BIM model was used to integrate the project schedule and the 3D model of the project, and four different execution patterns were defined in an assumed rectangular workspace for each activity (Figure 2.17). The crews were designed to move in the simulation according to the patterns within a task execution and since the starting point of execution can be within four different choices in a rectangular area, a total of sixteen alternatives can be chosen in the proposed system. Finally, the system evaluates the alternatives and patterns, and a case with minimum conflict among the activities is selected for the corresponding crew.

In the studies referenced, work crews follow fixed movement patterns without the ability to independently assess environmental conditions and make decisions based on those assessments. Furthermore, the methods used did not account for uncertainties, resulting in simulations that reflect a deterministic scenario. This proposed method in the current thesis advances the field by enabling agents to make decisions based on dynamic conditions that arise from simulation results. This

approach helps planners identify time-space conflicts and assess their impact on project performance.

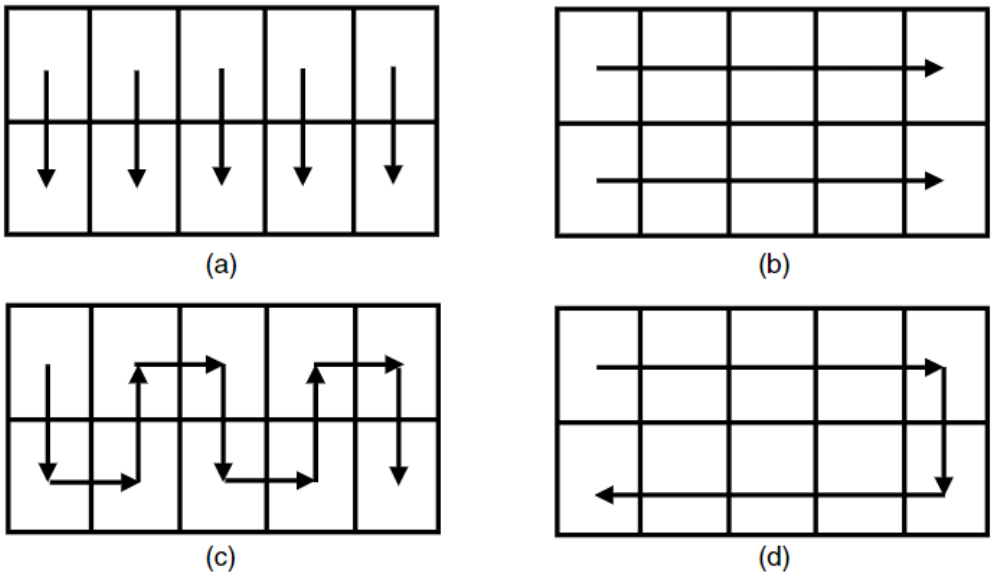


Figure 2.17. Predefined execution patterns (Mirzaei et al., 2018)

Messi et al. (2022) developed a simulation method that combines static and dynamic conflicts and evaluates the conflict severity by a Bayesian Network. The simulation is implemented in the Unity3D engine by integrating the work schedule and BIM model. They compared the proposed method with commercially available tools including Autodesk Navisworks and Bentley Synchro 4D and demonstrated the capability of the method in the detection of critical conflicts. Additionally, the possibility of conflicts caused by falling parts of the model elements is also addressed using the gravity feature of game engine objects. This feature is useful to take into account safety considerations.

Dashti et al. (2021) propose a hybrid simulation method that relies on BIM, Discrete Event Simulation (DES), and a Rapidly exploring Random Tree (RRT) path-finding algorithm. There is an “if-then” mechanism that is responsible for quantifying the effect of the conflict on productivity, priority, and executability. They implemented the method as an add-in to the Autodesk Navisworks that landed on the C# .Net framework. Using the add-in the required space for each simulation element is

generated in Autodesk AutoCAD 3D and imported to the model. The conflicts are assessed following three methods Bounding Spheres (BS), Axis Aligned Bounding Box (AABB), and Oriented Bounding Box (OBB) (Figure 2.18).

Tak et al. (2021) proposed a 4D BIM-based simulation method to detect space congestion among the mobile cranes in a construction project at both macro and micro levels which is implemented on Autodesk Navisworks API. Alongside conflict detection, the method can help decision-makers identify hazardous situations in the site by performing lift simulations. The system gathers geometry, schedule, cost, and equipment data and uses a central database to support the analysis.

Fathi et al. (2023) developed a method to manage space congestion using DES and the informed-RPT path planning algorithm in the Unity gaming engine (Figure 2.19). The system detects the congestion cases and automatically resolves them during the simulation. They tested the method on the earthwork processes in a construction project to demonstrate the capability of finding the shortest path for each truck by avoiding space conflicts.

A multi-domain conflict detection method for unmanned aerial vehicles (UAVs) was developed by Junjie et al. (2023). The method identifies three types of conflicts including space conflicts, time-space coupling conflicts, and time-space-frequency coupling conflicts. Decoupling the conflicts into manageable parts is used as the resolution policy for congestion cases. The simulation results indicated fewer conflicts compared to the conventional approaches.

Table 2.3 provides a comparison between a few previous studies and the current one, aiming to highlight existing gaps and demonstrate the innovative aspects of the proposed method for addressing these limitations. The table outlines the methods used and compares them across several criteria, including in-path conflict support, model environment, conflict avoidance capability, dynamic simulation support, inclusion of uncertainty through probabilistic parameter variation, crew behavior modeling support, and time-space conflict detection.

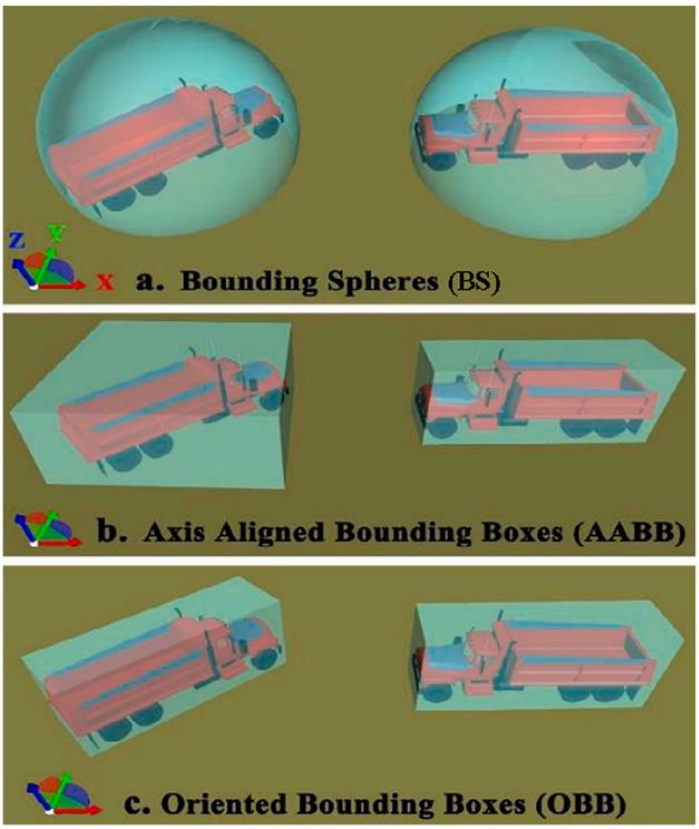


Figure 2.18. Three bounding volume methods for space conflict analysis (Dashti et al., 2021)

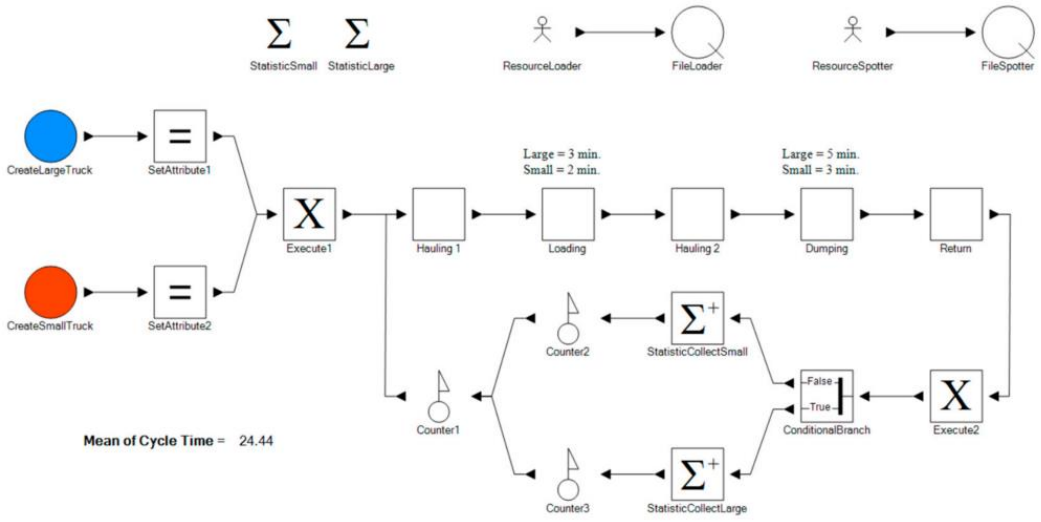


Figure 2.19. The DES developed for earthwork processes (Fathi et al., 2023)

Table 2.3 A comparison between the previous and present studies

Reference	In-path conflict detection	Modeling environment	Time-space conflict detection	Conflict avoidance mechanism	Dynamic simulation	Probabilistic approach	Crew behavior modeling
Thabet & Beliveau (1994)	No	CAD	No	No	No	No	By three class of attributes
Thabet & Bealivean (1997)	No	CAD	No	No	No	No	By a database structure
Riley & Sanvido (1997)	Yes	CAD for testing the suggested process method	No	No	No	No	By defining space behaviors
Guo (2002)	Yes	CAD	Yes	No	No	No	By differentiating space types
Akinci et al. (2002)	No	4D CAD (4D TSConAn)	Yes	No	Yes	No	By differentiating space types
Mallasi (2006)	No	4D CAD (PECASO)	Yes	No	Yes	No	By defining execution patterns
Tantisevi & Akinci (2007)	No	4D CAD	Yes	No	Yes	No	By modeling constituting parts (applied for cranes)
Hu & Zhang (2011)	No	4D BIM	Yes	No	Yes	Yes	No
Mawlana et al. (2015)	-	4D + DES	Yes	No	Yes	Yes	No
Mirzaei et al. (2018)	No	4D BIM	Yes	No	Yes	No	By execution patterns
Dashti et al. (2021)	Yes	BIM + DES	Yes	Yes	Yes	No	No
Tak et al. (2021)	Yes	4D BIM	Yes	Yes	Yes	No	No
Messi et al. (2022)	No	4D BIM	Yes	Yes	Yes	No	No
This study	Yes	4D BIM + ABS	Yes	Yes	Yes	Yes	By agents via their behavior patterns

CHAPTER 3

BIM-INTEGRATED ABS METHOD FOR QUANTIFYING TIME-SPACE CONFLICTS

This chapter introduces the proposed method on the basis of method architecture. Each step in the architecture is explained with supportive information to clarify the method structure and provide a comprehensive understanding of the process. It should be noted that this chapter serves as a general outline of the methodology; therefore, the details of the method are provided in Chapter 4, where the implementations demonstrate the method in practice, offering a comprehensive understanding of its application and effectiveness.

3.1 Method Architecture

In this section, the proposed BIM-integrated ABS method is thoroughly explained based on a system architecture comprising four main segments, as illustrated in Figure 3.1.

This architecture provides a roadmap for achieving a BIM-ABS from a 4D model and user inputs, enabling the quantification of time-space conflicts, tracking impacts on project performance, and subsequently informing the crashing analyses.

3.1.1 Data Extraction, Agent Pool, and Inputs

This step outlines the parameters that the user provides, including an agent pool and embedded data extracted from a 4D model. To set up a simulation, the user should specify a new crew size based on different crashing alternatives, and define simulation parameters including crew speed, time scale as an indicator of simulation

speed, number of iterations, and corresponding probability distributions to take uncertainty into account. Additionally, an agent pool comprising agents representing crews with various behavior patterns is created and stored in an external database. The behavior patterns of an agent include demanded space, occupation attributes, state, speed, size, and more. Users can customize these parameters based on project conditions and construction methods.

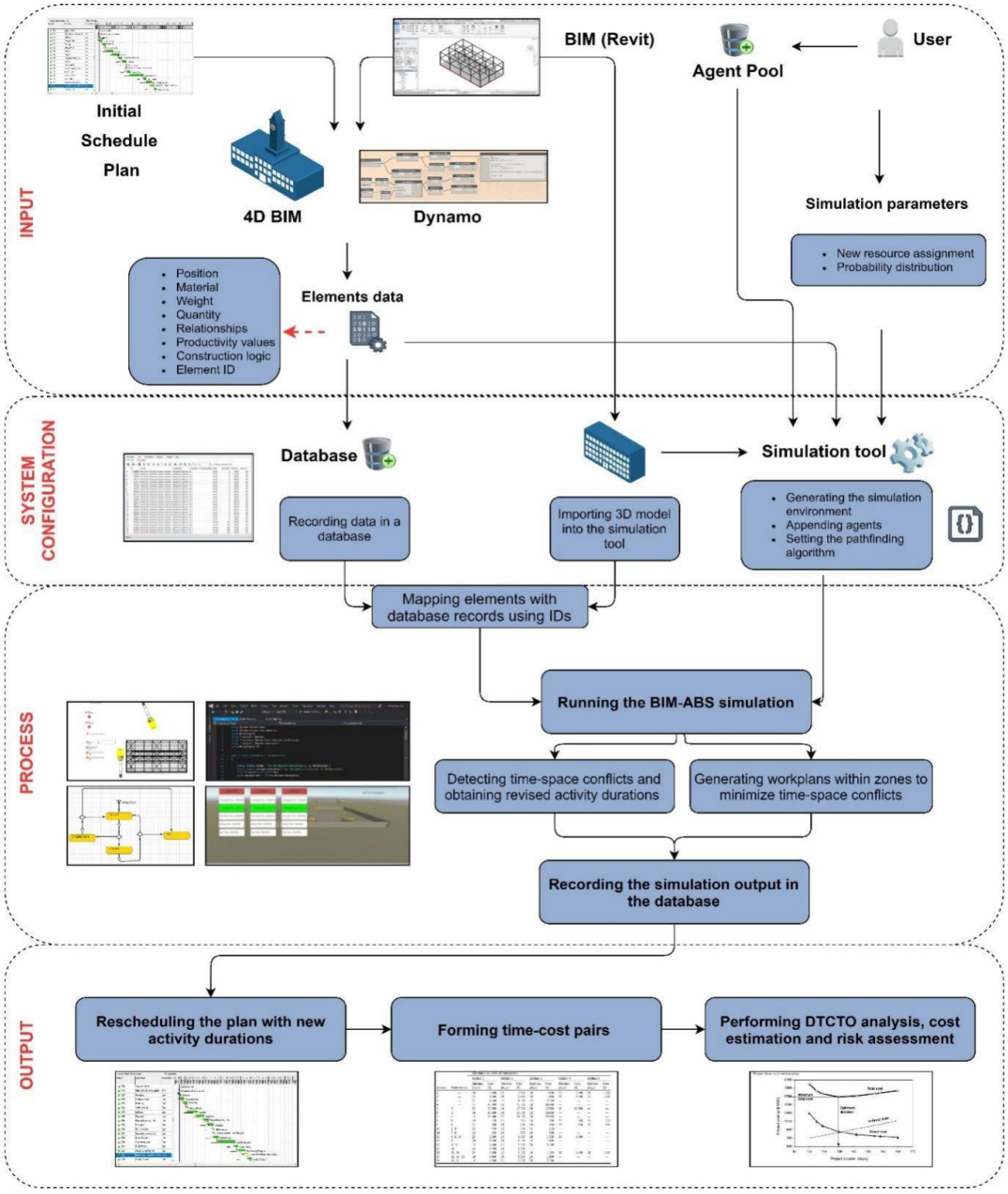


Figure 3.1. The proposed method architecture

The agents are then added to the simulation according to the project schedule derived from the 4D model, new resource allocation, and construction logic. Moreover, a visual programming tool is used to extract and organize element data, such as spatial information, material, quantities, and productivity values from the BIM model for use in the simulation. Each element in the model has a unique ID to help map it with corresponding data stored in the external database. The associated code for the mentioned processes is outlined as the following pseudocode (Figure 3.2).

```
Function add_to_db(data) {
    // Construct and execute the SQL insert statement
    SQL "INSERT INTO table_name (column1, column2, column3, ...)
         VALUES (data.value1, data.value2, data.value3, ...)";
}

For Each element in document:
    dbRow = {}
    For Each parameter in element:
        dbRow[parameter.name] = parameter.value
    End For
    add_to_db(dbRow)
End For
```

Figure 3.2. A pseudocode for extracting BIM model data and inserting to the database table

3.1.2 System Configuration

In this part, the simulation parameters are set, agents are added, and the BIM model is used to create the simulation environment. A database is also used to store the data extracted to be used by the simulation during each run and to report the outputs for the crash analyses. A simulation platform that supports BIM-ABS in an object-based environment should be established to implement this method.

3.1.2.1 A* AI-based Pathfinding Algorithm

A* is a generic search method that can be used to solve several issues, including pathfinding. The A* algorithm constantly assesses the most potential undiscovered spots for pathfinding tasks. The method ends if the destination is the studied position; if not, it logs all surrounding areas for additional exploration (Cui & Shi, 2010). The algorithm is based on the calculation of the cost associated with each adjacent point and selecting the one with the minimum cost. The mentioned cost relies on two factors: the exact cost of the possible next point and the forecasted cost until the destination point from the possible next point (Equation 3). The pseudocode of the algorithm is illustrated in Figure 3.2.

$$F = G + H \qquad \text{Equation 3}$$

The A* algorithm employs a heuristic approach. While it does not guarantee the absolute best result, it ensures that a solution is found within a reasonable time frame. Typically, A* efficiently and swiftly arrives at an optimal solution (Aydoğan & Atay, 2021).

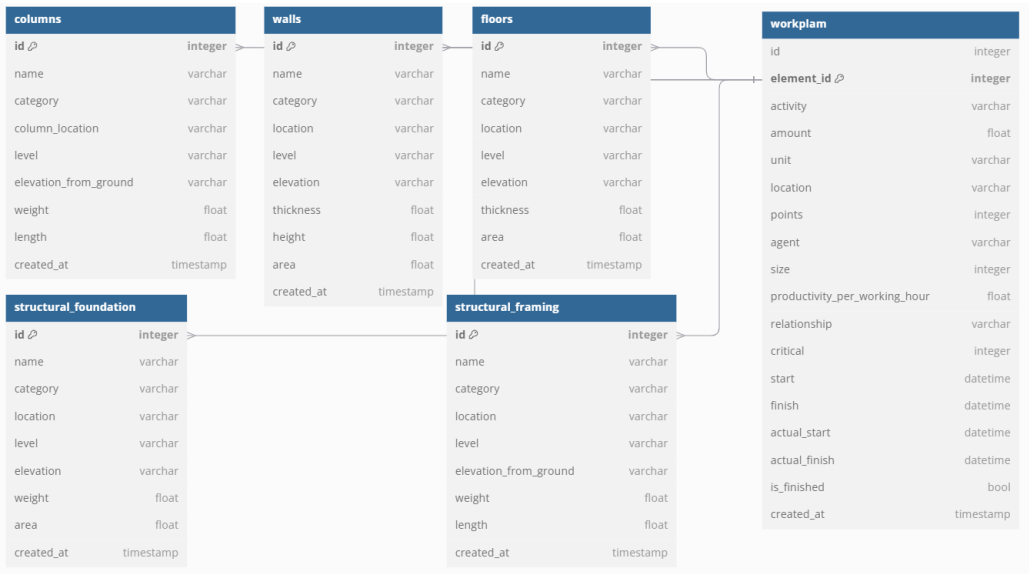


Figure 3.3. The database structure to store model element information

```
1.  Add the starting node to the open list.
2.  Repeat the following steps:
    a.  Look for the node which has the lowest
        f on the open list. Refer to this node
        as the current node.
    b.  Switch it to the closed list.
    c.  For each reachable node from the current
        node
         i.  If it is on the closed list, ignore
             it.
        ii.  If it isn't on the open list, add it
             to the open list. Make the current
             node the parent of this node. Record
             the f, g, and h value of this node.
       iii.  If it is on the open list already,
             check to see if this is a better
             path. If so, change its parent to the
             current node, and recalculate the f
             and g value.
    d.  Stop when
         i.  Add the target node to the closed
             list.
        ii.  Fail to find the target node, and the
             open list is empty.
3.  Tracing backwards from the target node to the
    starting node. That is your path.
```

Figure 3.4. The pseudocode of A* algorithm (Cui & Shi, 2010)

Unlike the pre-defined crew movement patterns used in previous studies, the pathfinding algorithm is used to support the natural movement of agents and obstacle detection in the environment. This algorithm ensures agents move through the simulation space realistically, considering dynamic changes and obstacles, making the simulation results more accurate and reliable. The pathfinding algorithm also allows for the adaptive behavior of the agents to model construction processes and potential issues more precisely.

3.1.2.2 Conflict Matrix

To differentiate the space conflicts causing constructability issues from workable congestions, a conflict matrix is defined (Figure 3.5). This matrix consists of ones and zeros indicating workability and unworkability respectively and is checked instantly following a space conflict. The latter case signifies the safety issues, logical

problems, or practical constraints that may arise when two or more agents operate within the same space. The productivity of congested but workable agents is revised according to conflict severity. But for the unworkable congested agents, a priority-based mechanism is suggested, that gives priority to the activities placed on the critical path. Provided that more than one activity is placed on the critical path, the agent with lower progress is prior to the other. This behavior can also be altered based on user preference.

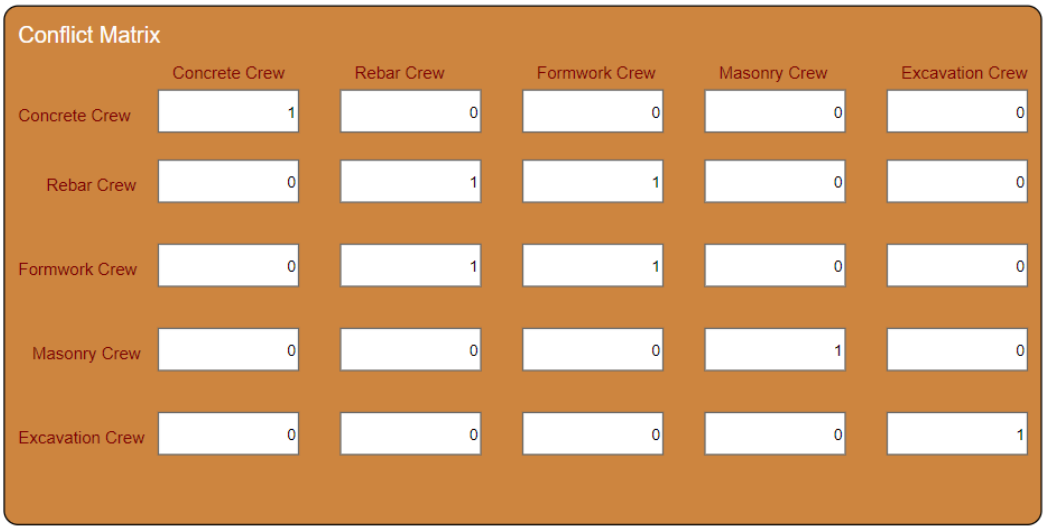


Figure 3.5. An example conflict matrix implemented in Anylogic

3.1.3 Simulation Run Process

In case the simulation parameters are probabilistic, it necessitates multiple runs to establish the empirical distribution functions. Given the independent nature of the agents within the simulation, a dynamic conflict avoidance mechanism is employed to model the crews' behavior within a working zone. These zones can be either an entire floor or a part of a larger floor. At each time step, a multicriteria function is used to generate an in-zone execution logic by considering both the positions of the agents and the proximity of the next possible working location, as described by Equation 4. The first criterion ensures that agents minimize or avoid space conflicts with others, while the second criterion takes into account the crews' tendency to work

close to their previous location, minimizing traffic due to the equipment they carry and the need to reset their material storage location.

$$F = C_1 M_1 + C_2 M_2 \qquad \text{Equation 4}$$

where M_1 is the scaled score assigned to the next probable destination point based on its distance from the previous point, M_2 is the scaled score assigned to the next probable destination point based on its distance from other agents, and $C_1$and C_2 are the impact factors of M_1 and M_2 respectively (with $(0 \leq C \leq 1)$). For example, this function for the mobile crane and steel structure installation agents can be defined in four stages:

I) The system initially nominates a set of points that won't lead to conflicts between agents. This is guided by a rule system or schedule plan that ensures logical sequencing of tasks. For instance, before installing a beam, the underlying columns must be installed first.

II) Once elements are installed, they are marked as previous points for future calculations (illustrated in

Figure 3.6). The agent then goes to the storage area to pick up the necessary steel materials and transports them to the next destination.

III) A set number of points (typically four, but this can be adjusted) closest to the previous point are identified as potential next destinations. For each of these points, M_1 is computed based on their distance from the previous point (see Equation 5). This helps determine the most efficient next step in terms of distance and effort required for the agent.

IV) The system dynamically identifies the locations of tasks being performed by other agents and measures the distance between these tasks and the nominated points (illustrated in Figure 3.7). With two agents in the zone, the highest M_2 score is assigned to the point farthest from the other agent to minimize interference. With more than two agents, the point that is equidistant from the other agents receives a better score, as calculated using Equation 6, Equation 7, and Equation 8. This ensures

a balanced distribution of agents across the workspace and prevents clustering, leading to more efficient operations. Table 3.1 provides an example calculation for the scenario in Figure 3.7, considering three agents working in the same zone, illustrating how the scoring system optimizes agent movements.

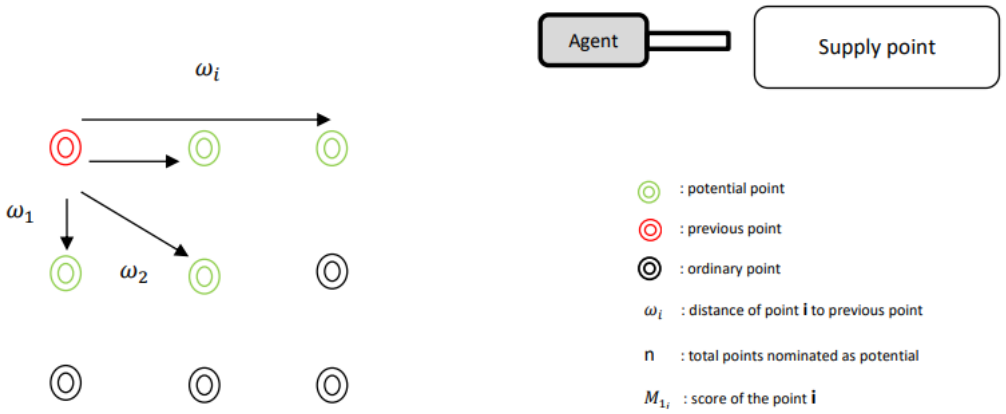


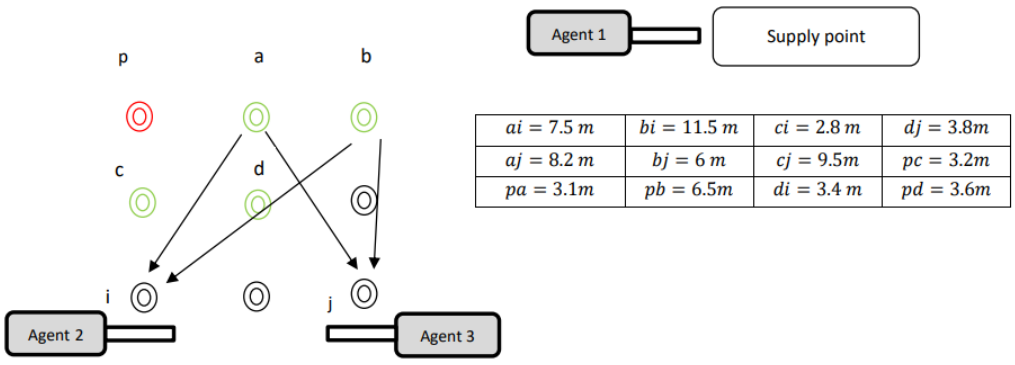
Figure 3.6. Nominating next destination points for M_1 calculation

$$M_{1_i} = \frac{w_i}{\sum_1^n w_i} \times 4 \qquad \text{Equation 5}$$

$$dev_a = max(|ai - avg(ai, aj)|, |aj - avg(ai, aj)|, \dots)) \qquad \text{Equation 6}$$

$$dev_b = max(|bi - avg(bi, bj)|, |bj - avg(bi, bj)|, \dots) \qquad \text{Equation 7}$$

$$M_{2_i} = \frac{dev_i}{\sum dev} \times 4 \qquad \text{Equation 8}$$



$ai = 7.5\ m$	$bi = 11.5\ m$	$ci = 2.8\ m$	$dj = 3.8m$
$aj = 8.2\ m$	$bj = 6\ m$	$cj = 9.5m$	$pc = 3.2m$
$pa = 3.1m$	$pb = 6.5m$	$di = 3.4\ m$	$pd = 3.6m$

Figure 3.7. Calculating other agents' locations to obtain M_2

Table 3.1 Calculating M1 & M2 for a sample project

M_1			M_2				**Function value**
$C_1 = 0.5$			$C_2 = 0.5$				$C_1M_1 + C_2M_2$
vector	magnitude	score	point/vector		magnitude	score	
pa	3.1	$\frac{3.1}{\sum_1^n \omega_i = 16.4} \times 4 = 0.75$	a	ai	7.5	$\frac{dev_a}{\sum dev} \times 4 = \frac{1.4}{6.65} = 0.53$	0.48
				aj	8.2		
pb	6.5	$\frac{6.5}{16.4} \times 4 = 1.58$	b	bi	11.5	$\frac{dev_b}{\sum dev} \times 4 = \frac{11}{6.65} = 1.65$	1.61
				bj	6		
pc	3.2	0.78	c	ci	2.8	2.01	1.39
				cj	9.5		
pd	3.6	0.87	d	di	3.4	0.12	0.50
				dj	3.8		

Agents become less productive when they have to work in a shared workspace with other agents at the same time (Figure 3.8). In Equation 8, the coefficient r represents this effect. It calculates the ratio of the intersecting area to the total needed space. Going even beyond when agents share the same workspace, their productivity is influenced by the degree of overlap in their work areas (see Equation 9). Due to the conflicts caused by this overlap, there are delays and lower productivity. Using the ratio of the overlapping area to the total area required for the agents' duties, the coefficient r provides a quantitative measure of this occurrence. Greater overlap is correlated with a higher r value, which suggests increased interactions and a steeper drop in productivity. By using this coefficient in productivity evaluations, it is possible to simulate and foresee the effects of shared workspaces on the agents' overall productivity.

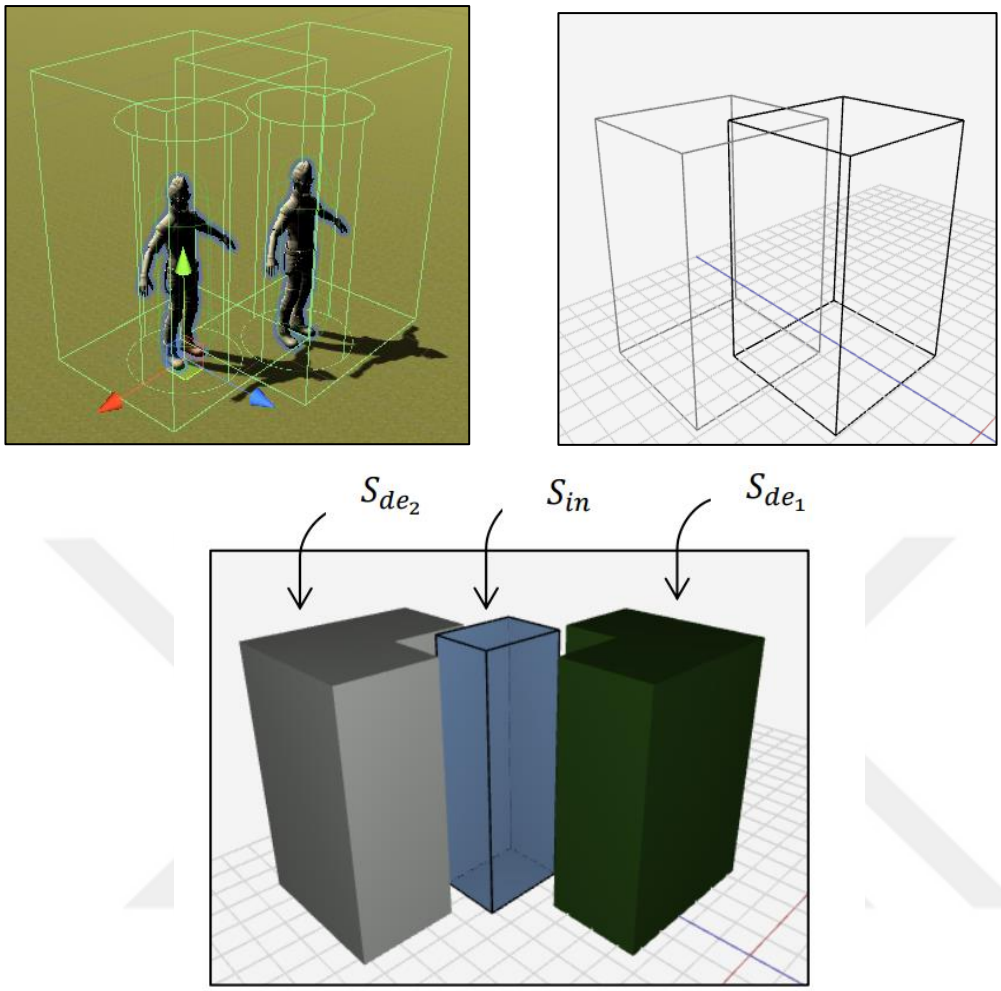


Figure 3.8. Determining the overlapping and required areas during a time-space conflict

$$r = \frac{S_{in}}{S_{de}} \qquad \text{Equation 9}$$

Where, S_{in} = intersected space among the agents; S_{de}= demanded space for each agent; and $(0 \leq r \leq 1)$.

$$p_r = p_0 \times (1 - r) \qquad \text{Equation 10}$$

Where, p_r= revised productivity value; and p_0 = initial productivity value

The variable r acts as an indicator of conflict severity and is recorded as an output of each simulation. In cases where agents can't perform their tasks simultaneously,

the agent (crew) in charge of a critical path activity is given precedence to use the space. This strategy aims to mitigate the impact of time-space conflicts on the project's duration by leveraging activity floats. Finally, the results of the simulations, encompassing quantified time-space conflicts, revised activities, and generated workplans, are documented in a database for subsequent crash analysis.

3.1.4 Simulation Output

The simulation data needs to be analyzed to develop crashing modes and assess time-cost trade-offs and associated risks. For each crashing alternative, the initial project schedule is updated to address identified time-space conflicts. This process results in a revised project schedule or an empirical distribution function when probabilistic parameters are applied. Consequently, each crashing alternative produces a time-cost pair, providing a quantitative evaluation of the project's duration along with the corresponding costs.

CHAPTER 4

CASE STUDIES

This chapter covers the implementation of the proposed method in two different software packages. Although each has distinct benefits and limitations, both implementations reveal the potential benefits of the method on decision-making in crashing schedule plans. The results of both implementations are analyzed and appraised alongside the DTCTO analysis.

4.1 Method Implementation in Gaming Engine

The BIM-integrated ABS method is implemented in an actual warehouse project involving the installation of steel structures. In this project, teams represented by agents are responsible for tasks such as lifting, transporting to the designated location, and installing structural steels. Mobile cranes are used for lifting and transporting, while erection crews handle the installation. The dynamic and complex nature of mobile crane operations poses challenges for conflict assessment with traditional methods, which often lack the necessary flexibility.

This warehouse project serves as a practical demonstration of the proposed method. It illustrates how combining BIM with ABS can enhance the planning and execution of construction tasks. This approach allows for a more detailed and accurate simulation of the construction process, enabling real-time identification and mitigation of conflicts. By offering a more adaptable and responsive method for conflict assessment, it improves both the efficiency and safety of the steel structure installation process.

4.1.1 BIM Model Creation and Exporting to Game Engine

For the first step, a structural BIM model for the mentioned project is generated in Autodesk Revit (Figure 4.1). Since the project only involves steel installation with the corresponding crew, steps like integrating the initial project schedule into BIM, creating the agent pool, and revising the schedule are excluded. The simulation environment is built using the Unity3D gaming engine, which is responsible for providing a dynamic aspect to the simulation and supporting collision detection through a virtual sensory system.

Given that game engines are typically not used for operational simulations, a custom simulation system is designed. This system is designed using C# programming language and the .Net framework, working in conjunction with the game engine to create the simulation and its environment. The 3D model from Revit is imported into Unity3D via an add-in called Walk-Through-3D, which converts the Revit file into a model compatible with the game engine (Figure 4.2). To facilitate agent movement and obstacle detection, the simulation uses an AI pathfinding algorithm known as A*. Unity provides a pathfinding mechanism based on the A* algorithm called NavMesh. This mechanism allows the agent to find the way in the walkable area through the generated mesh considering the obstacles in the path (Figure 4.3).

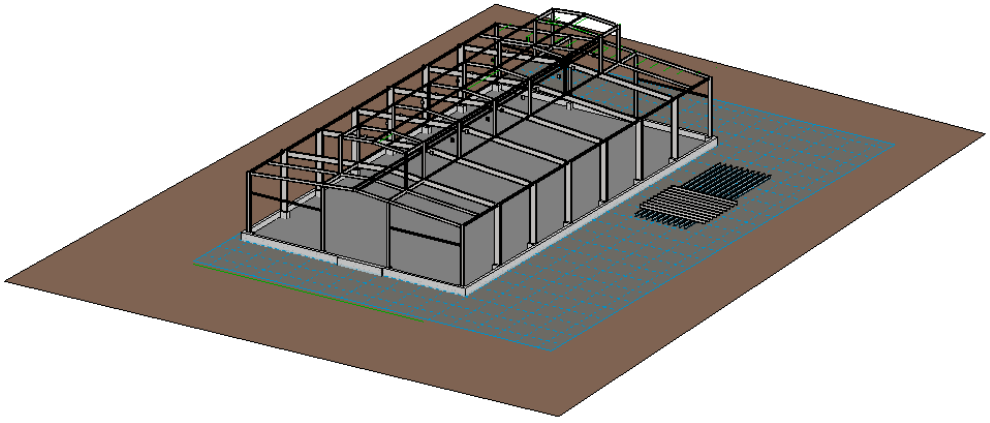


Figure 4.1. BIM-based model of the warehouse project

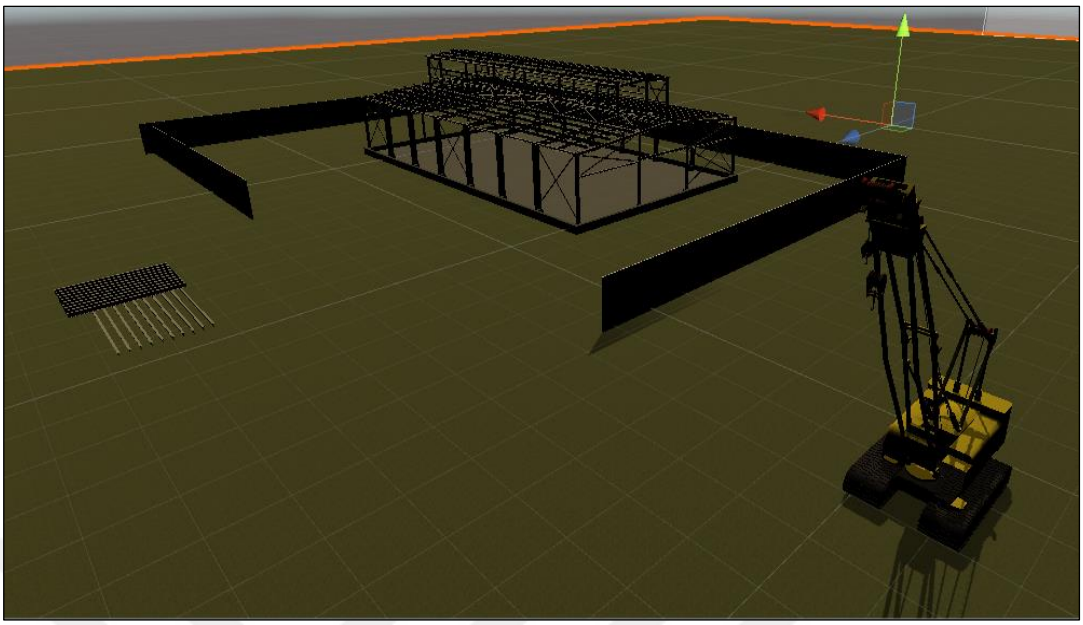


Figure 4.2. Imported model in the game engine environment

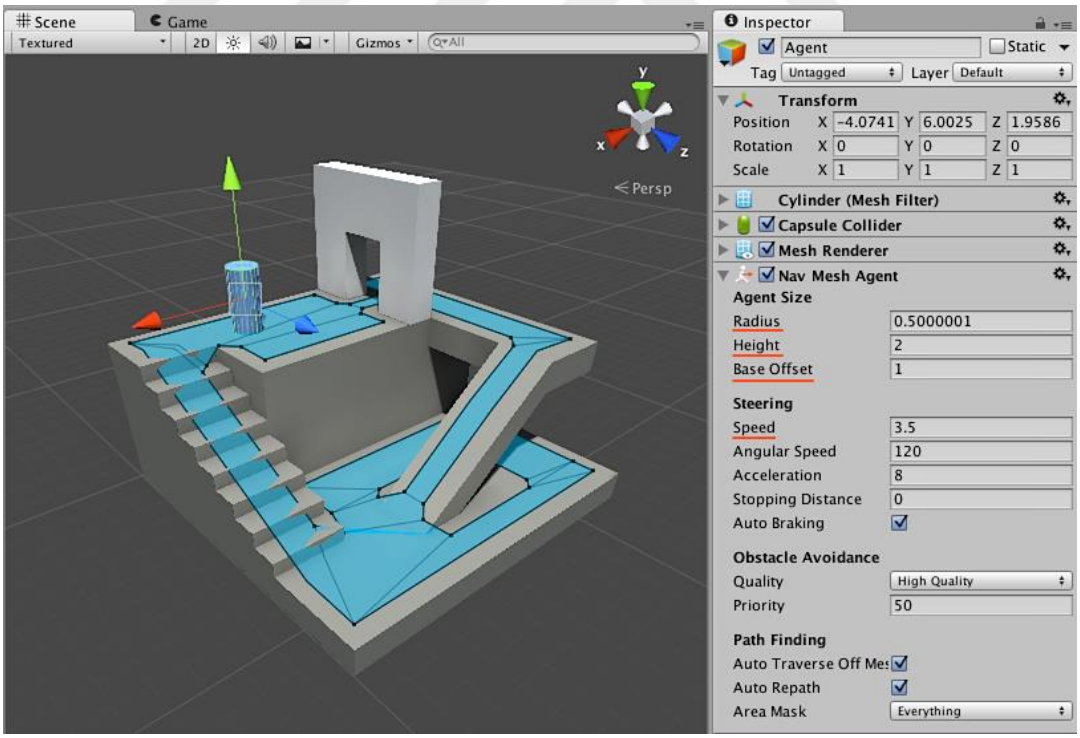


Figure 4.3. NavMesh mechanism in Unity (adapted from Unity documentation)

4.1.2 Collision Detection Mechanism in Game Engine

Moreover, there is a physics system in Unity that provides collision detection and physical simulation. This system is used to model the interaction of the agents and space conflict detection by intersecting the spaces through the mentioned system (Figure 3.8). Once a collision is detected, a method is triggered to calculate the r coefficient within a C# function. Accordingly, the productivity value for the agents is revised and recorded on a database for further analysis. The associated function that is named onTriggerEnter is presented in Figure 4.4. This function is embedded in the game object (the agent) and is triggered once another object, which is called a collider, is within the collision box representing the working space of the agent.

```
public void OnTriggerEnter(Collider other)
    {
        if (agents.Contains(other.gameObject))
        {
            OriginalState = State;
          if (OriginalState != state.Idle &&
          other.GetComponent<Agent>().State != state.Idle &&
          other.GetComponent<Agent>().State != state.Congestion &&
          gameObject.GetComponent<NavMeshAgent>().avoidancePriority <
          other.GetComponent<NavMeshAgent>().avoidancePriority)
                {
                    calculateRcoefficient();
                    otherTriggered = other;
                    State = state.Congestion;
                    proximityOnPath = true;
                }
            }

            else
            {
                proximityOnPath = false;
            }
    }
```

Figure 4.4. Collision detection script in C#

By collision detection, the function checks the type of collider if it is an agent or a static obstacle. Once an agent is detected the state of the collided agents is revised, and the coefficient r is calculated, then the productivity of the agents is updated accordingly. This effect is reversed once the collision is cleared through another function as shown in Figure 4.5.

```
public void OnTriggerExit(Collider other)
{
    proximityOnPath = false;
    if (agents.Contains(other.gameObject))
    {
        if (State == state.Congestion)
        {
            State = OriginalState;
        }
    }
}
```

Figure 4.5. Clearance of the collision detection function

4.1.3 Agent State and Data Collection

Each agent in the simulation environment gets a state based on its perception of the environment or interactions with other agents. In this respect, a state chart outlining the potential states of each agent is established during their creation (Figure 4.6). It should be noted that in the case project involving mobile cranes, safety protocols prevent the agents from operating simultaneously. Consequently, in situations where congestion leads to a breach of safety distances between agents, the agent with lower priority pauses its operation until the others have finished, instead of calculating the r factor and the corresponding reduction in productivity. Furthermore, to support further analyses, the simulation, data is extracted from the model using a visual programming tool named Dynamo, which extracts information from the BIM model and stores it in a relational serverless database called SQLite (Figure 4.7). During

this process, the elements' names are replaced with their IDs to simplify the mapping process during the simulation.

Dynamo is an open-source VPL that enhances the user interaction with the model in several environments including Autodesk's BIM tool called Revit. It has a specific scripting language and supports Python programming to provide more access to the tool API. In the current study, it is used to access the model elements data, manipulate the data based on the needs, and get connected to the SQLite database to perform CRUD operations. Figure 4.7 illustrates the mentioned steps through the

Dynamo VLP and Figure 4.8 shows the Python code for inserting the data into the database table and Figure 4.9 illustrates the database table extracted using Dynamo. The parameters needed for the current case project are element ID, name, category, type, volume, length, level, and location.

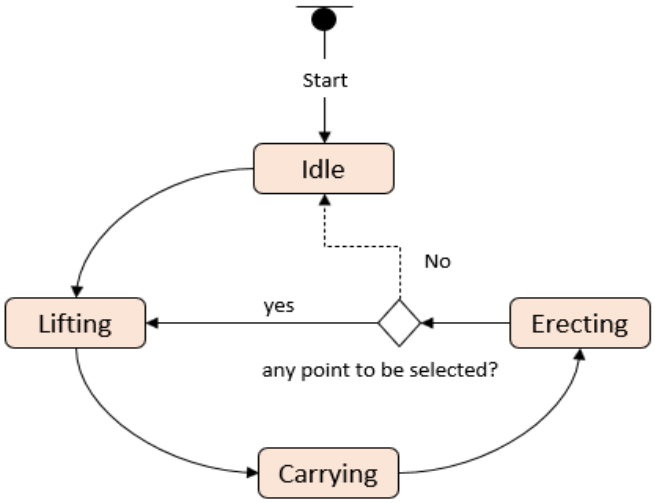


Figure 4.6. Mobile crane agent state chart

A user interface (UI) is created to gather user preferences regarding the simulation parameters, as illustrated in Figure 4.10. While the physical characteristics and behavior patterns of the agents were predefined, users can choose to modify these parameters by defining probability distributions or by using deterministic values. When the user initiates the system, the simulation runs for the specified number of

cycles, and the results are recorded in the SQLite database for each crew size, representing the various crashing alternatives. The Unity pathfinding algorithm determines the agents' movement based on their embedded behaviors and a conflict avoidance function that governs their execution patterns within specific zones. In other words, an additional layer is added to the pathfinding mechanism that takes into account the multicriteria function (Equation 3) to increase the fidelity of the simulation behavior in the crew traffic.

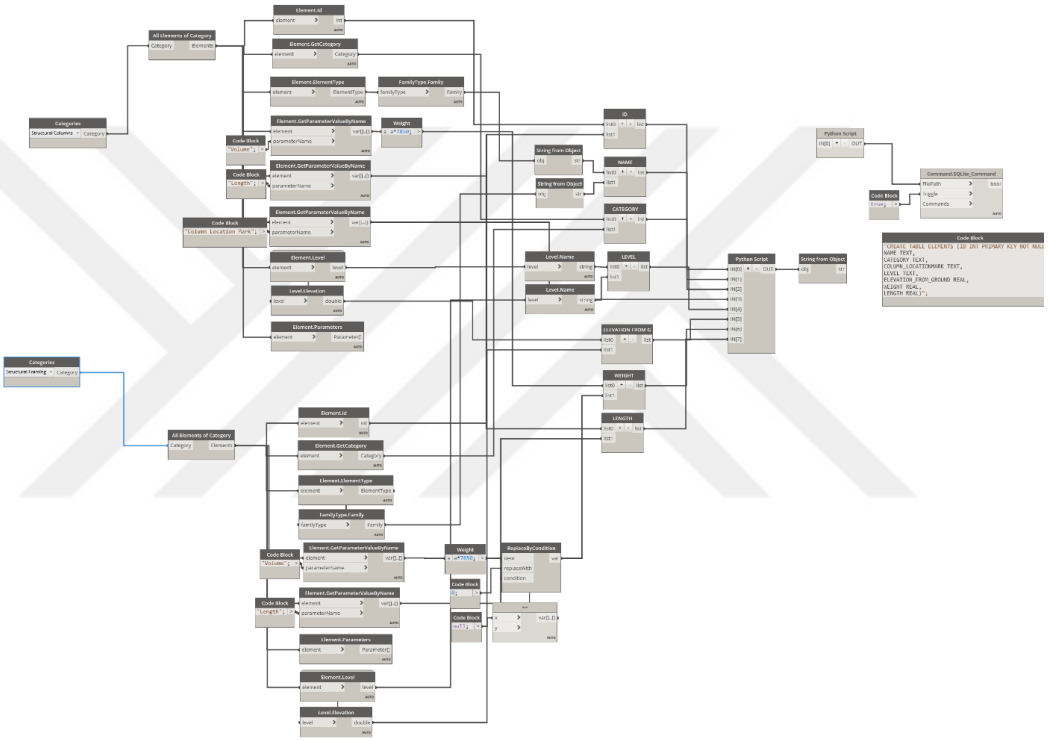


Figure 4.7. Extracting BIM model data using Dynamo VLP

```
# The inputs to this node will be stored as a list in the IN variables.
ID = IN[0]
NAME = IN[1]
CATEGORY=IN[2]
COLUMNL=IN[3]
LEVEL=IN[4]
ELEVATION=IN[5]
WEIGHT=IN[6]
LENGTH=IN[7]
a=[]
# Place your code below this line
for i in range(len(ID)):
    if i< len(COLUMNL) and COLUMNL[i] != "":
        a.append("INSERT INTO ELEMENTS
        (ID,NAME,CATEGORY,COLUMN_LOCATIONMARK,LEVEL,ELEVATION_FROM_GROUND,WEIGHT,L
        ENGTH) VALUES ("+str(ID[i])+",'"+NAME[i]+"','"+str(CATEGORY[i])+"','"+str
        (COLUMNL[i])+"','"+str(LEVEL[i])+"',"+str(round(ELEVATION[i],2))+","+str
        (round(WEIGHT[i],2))+","+str(round(LENGTH[i],2))+")")
    else:
        a.append("INSERT INTO ELEMENTS
        (ID,NAME,CATEGORY,COLUMN_LOCATIONMARK,LEVEL,ELEVATION_FROM_GROUND,WEIGHT,L
        ENGTH) VALUES ("+str(ID[i])+",'"+NAME[i]+"','"+str(CATEGORY[i])+"','Not
        Column Location Mark','"+str(LEVEL[i])+"',"+str(round(ELEVATION[i],2))
        +","+str(round(WEIGHT[i],2))+","+str(round(LENGTH[i],2))+")")
# Assign your output to the OUT variable.
OUT = a
```

Figure 4.8. A Python code to insert the data into the database table

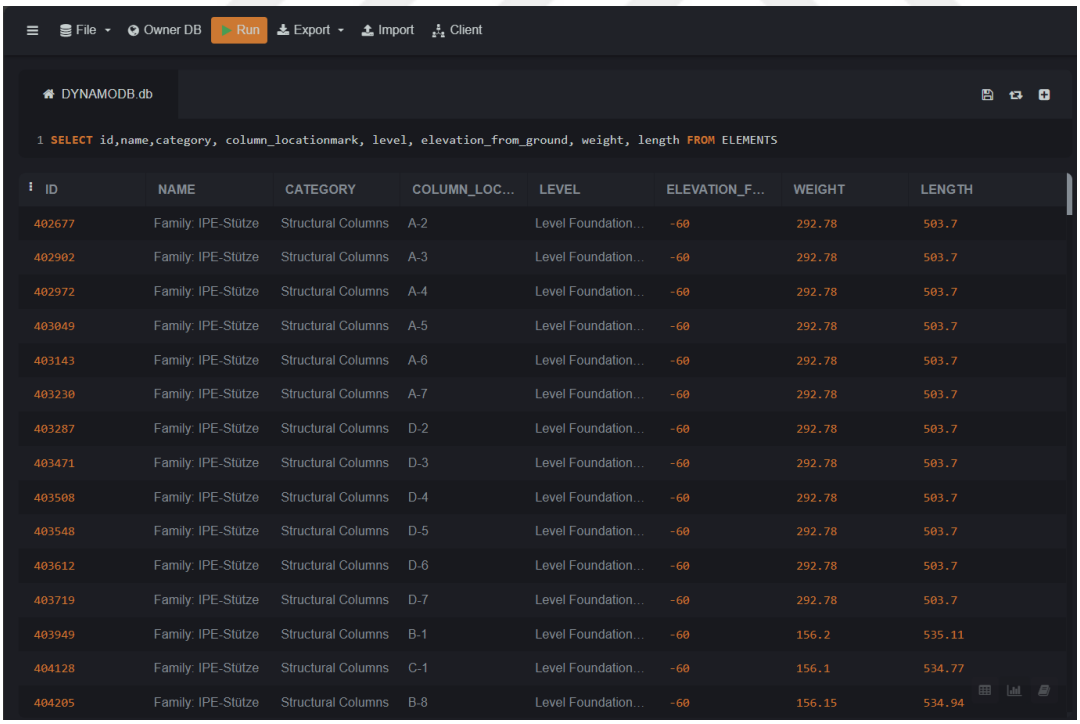


Figure 4.9. SQLite database filled with the BIM model data

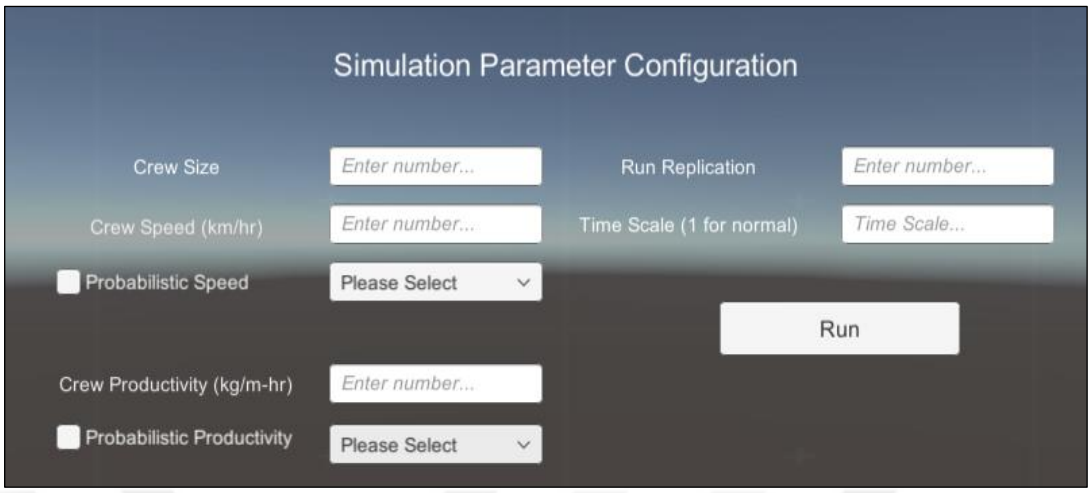


Figure 4.10. User interface (UI) for getting user inputs in Unity

4.1.4 Simulation Run

As depicted in Figure 3.7 from a plan view, agent 1 is going to choose its next destination point. Therefore, M_2 is calculated for points a, b, c, and d. These points are nominated from two under-execution points by agents 2 and 3. Points a and d are prioritized over c and b because they are equidistant from points i and j (currently being executed by agents 2 and 3, respectively). Both M_1 and M_2 are scaled to 4, and the multicriteria function assigns a rank up to 4 for each potential point. The point with the lowest rank is opted as the next element to be installed. In this example, considering points a, b, c, and d, point a is chosen as the next point. The coefficients C_1 and C_2, reflecting the impact of each score factor (M_1 and M_2) on the project, are based on user judgment and can be adjusted during the simulation's setup stage. The simulation then records the outcomes in the SQLite database for further analysis. The mentioned calculations are implemented in the Unity engine using C# script in an embedded function as shown in Figure 4.11.

Using the UI for the case project, the user input parameters including crew size, crew speed, productivity values in both deterministic and probabilistic forms, replication number, and time scale are set. The crews assumed to be telescopic mobile cranes

and corresponding installation teams, were defined based on technical features, engineering judgments, or assumptions. The crew speed in "pick & carry" mode was set using a triangular probability distribution with a lower limit of 3 km/hr, an upper limit of 8 km/hr, and a mode value of 5 km/hr. Similarly, the productivity rates for the structural steel installation crew were set using a triangular distribution with values of 35 kg/hr (lower limit), 85 kg/hr (upper limit), and 60 kg/hr (mode). The impact factors (C_1 and C_2) were both set to 0.5 (Figure 4.12).

The interaction of the agents and their appropriate behaviors are modeled using a mechanism in Unity called Coroutine. Coroutines are the methods allowing the separation of the tasks with the associated frames and providing control over the execution of the tasks. They are not providing multi threads but asynchronous operations and sequence of events within the run time. In other words, using Coroutines the execution of specific tasks waits for the completion of some other ones. This feature helps the model implementation of the agent behaviors through the associated interactions. The Coroutines encapsulate special methods in C# with a return type of the interface called IEnumerator which is a type of "System.Collections" namespace. To Implement a Coroutine, IEnumerator should include a return statement of "yield" as shown in a sample implementation in Figure 4.13. In the given code "WaitForSeconds" method stops the execution of the next operation until a specific time. An implementation of this method in the simulation is given in Figure 4.14.

The data previously stored in the database is retrieved using another C# script at the beginning of the simulation run. They are used to calculate the workload, execution logic, element mapping, and other needed data. For this purpose, a connection to the database is established first and the data fetching is implemented as presented in the Figure 4.15. Moreover, the simulation run results are recorded in another table in the database in order to support further analysis regarding the crashing alternatives.
In the first attempt, the simulation is run with deterministic parameters for one to six agents with 5 kilometers per hour maximum speed of agents, 60 kilograms per hour erection crew productivity, and impact factors (C_1 and C_2) 0.5 equal for both. The

simulation run results with the configurations and the associated task duration amounts are presented in Figure 4.16. Additionally, the reduction in task execution duration as the crew size increased is illustrated in Figure 4.17. This figure highlights how adding more agents influences the overall efficiency and speed of task completion. The data shows a clear trend of diminishing returns, where the rate of duration reduction decreases as more agents are added. Furthermore, Figure 4.18 presents a stacked chart of agent states, showcasing the distribution of time spent in various states by the agents under different crew sizes. This visualization provides insight into how agents' activities and idle times change with varying crew sizes.

For the next stage, the parameters are considered probabilistic, and simulation is run 50 times for each crew size, with an 8-hour workday, and the results are recorded in the database and transferred to an Excel sheet. Since the simulation is probabilistic, an empirical cumulative distribution function (ECDF) is generated as the output, showing the project duration for different crew sizes in a probabilistic format (Figure 4.19). For risk analysis, various percentiles from the distribution can be used to provide duration values for different crashing alternatives. Table 4.1 shows three percentiles: 10%, 50%, and 90% from the simulation runs. For example, 90% of the duration values are less than 158 working hours with a crew size of three. This analysis contributes to the decision-making process to estimate duration ranges for time and cost risk analyses.

```
List<Transform> ProbablePoints = new List<Transform>();
List<KeyValuePair<Transform, int>> DistanceScore = new List<KeyValuePair<Transform, int>>();
ProbablePoints = AgentDestinationPoints.Where(s => s.tag != "Busy").ToList().OrderBy(s =>
Vector3.Distance(FindPosition(PreviousPoint.transform), FindPosition(s))).Take(4).ToList();
int c = 4;
foreach (var item in ProbablePoints)
{
    DistanceScore.Add(new KeyValuePair<Transform, int>(item, c));
    c -= 1;
}
List<KeyValuePair<Transform, float>> ProximityScore = new List<KeyValuePair<Transform,
float>>();
for (int i = 0; i < ProbablePoints.Count; i++)
{
    List<float> distanceFromBase = new List<float>();
    for (int j = 0; j < otherAgents.Count; j++){
        if (counter == 1){
            distanceFromBase.Add(Vector3.Distance(FindPosition(ProbablePoints[i])          ,
otherAgents[j].GetComponent<Agent>().FindPosition(transformObject)));
        } else {
            if (otherAgents[j].GetComponent<Agent>().State != state.Idle)
            {
                distanceFromBase.Add(Vector3.Distance(FindPosition(ProbablePoints[i]),
otherAgents[j].GetComponent<Agent>().FindPosition(transformObject)));
            }}}
    float ProximityDeviation = 0;
    if (distanceFromBase.Count != 0)
    {
        ProximityDeviation = Math.Abs(distanceFromBase.OrderBy(s => s -
distanceFromBase.Average()).Last() - distanceFromBase.Average());
    }
    ProximityScore.Add(new KeyValuePair<Transform, float>(ProbablePoints[i],
ProximityDeviation));
}
float C1 = 0.5f; // C1 & C2 can be modified based on the user preference
float C2 = 0.5f;
float M1, M2;
List<KeyValuePair<Transform, float>> ObjectiveFunctionPairList = new
List<KeyValuePair<Transform, float>>();
foreach (var item in ProbablePoints)
{
    M1 = DistanceScore.Single(s => s.Key == item).Value;
    M2 = ProximityScore.Single(s => s.Key == item).Value;
    ObjectiveFunctionPairList.Add(new KeyValuePair<Transform, float>(item, C1 * M1 + C2 * M2));
}
```

Figure 4.11. Implementation of multicriteria function in C#

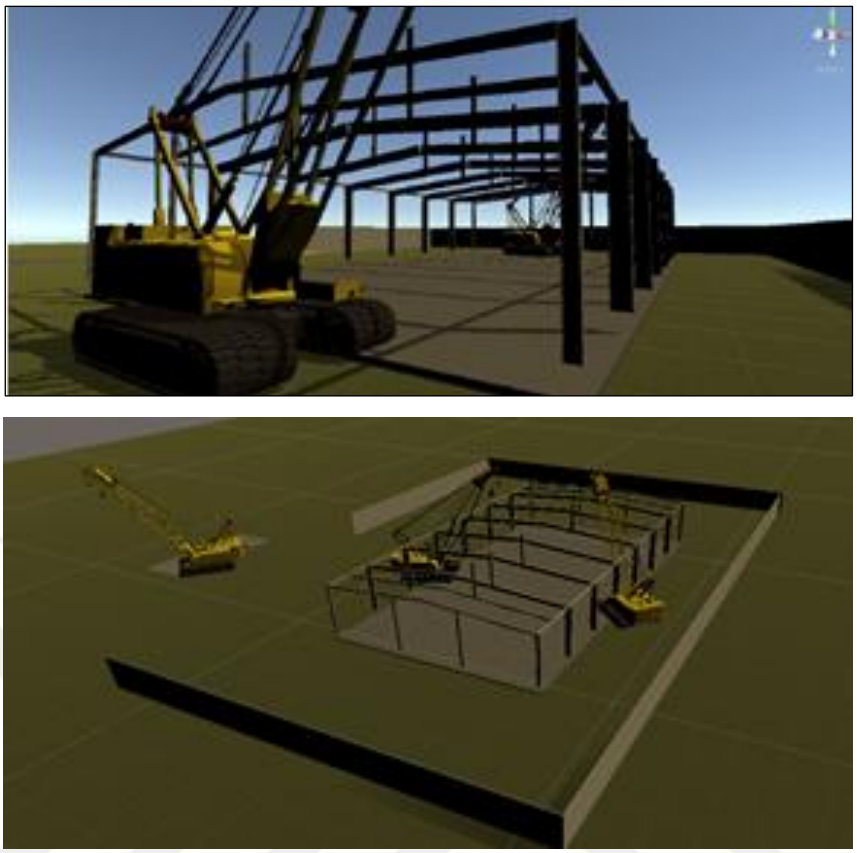


Figure 4.12. A simulation scene in Unity

```
IEnumerator DoCheck()
{
    for(;;)
    {
        if (ProximityCheck())
        {
            // Perform some action here
        }
        yield return new WaitForSeconds(.1f);
    }
}
```

Figure 4.13. An example Coroutine implementation in Unity (adapted from Unity manual)

```
IEnumerator executionLogic()
    {
        mesh = GetComponent<NavMeshAgent>();
        for (int i = 0; i < childrenObjects.Count; i++)
        {
            transformObject =
GameObject.Find(childrenObjects[i].gameObject.name).transform;
            agent.GetComponent<NavMeshAgent>().speed = agentSpeed;
            refVector =new
Vector3(transformObject.GetComponent<BoxCollider>().bounds.center.x,0f,
transformObject.GetComponent<BoxCollider>().bounds.center.z);
            mesh.SetDestination(refVector);
            yield return new WaitUntil(checkLocation);
            yield return new WaitUntil(checkProximity);
            yield return new WaitForSeconds(unloadTime);
            Back();
            yield return new WaitUntil(FinishLoad);
            yield return new WaitForSeconds(unloadTime);
        }
    }
```

Figure 4.14. An implementation of Coroutine in the simulation

```
public void readerDataFromDb(int GetID)
    {
        using (dbconn = new SqliteConnection(conn))
        {
            dbconn.Open(); //Open connection to the database.
            dbcmd = dbconn.CreateCommand();
            sqlQuery = "SELECT * " + "FROM ELEMENTS WHERE ID="+GetID;//table name
            dbcmd.CommandText = sqlQuery;
            IDataReader reader = dbcmd.ExecuteReader();
            while (reader.Read())
            {
                int ID = reader.GetInt32(0);
                SQLID = ID;
                string NAME = reader.GetString(1);
                SQLName = NAME;
                string CATEGORY = reader.GetString(2);
                SQLCategory = CATEGORY;
                string COLUMN_LOCATIONMARK = reader.GetString(3);
                SQLColumnL = COLUMN_LOCATIONMARK;
                string LEVEL = reader.GetString(4);
                SQLLEVEL = LEVEL;
                float ELEVATION_FROM_GROUND = reader.GetFloat(5);
                SQLELEVATION = ELEVATION_FROM_GROUND;
                float WEIGHT = reader.GetFloat(6);
                SQLWEIGHT = WEIGHT;
                float LENGTH = reader.GetFloat(7);
                SQLLENGTH = LENGTH;
                var USAGE = reader.GetValue(8);
                SQLUsage = USAGE.ToString();

            }
            reader.Close();
            dbcmd.Dispose();
            dbconn.Close();
        }
    }
```

Figure 4.15. C# script for retrieving data from database

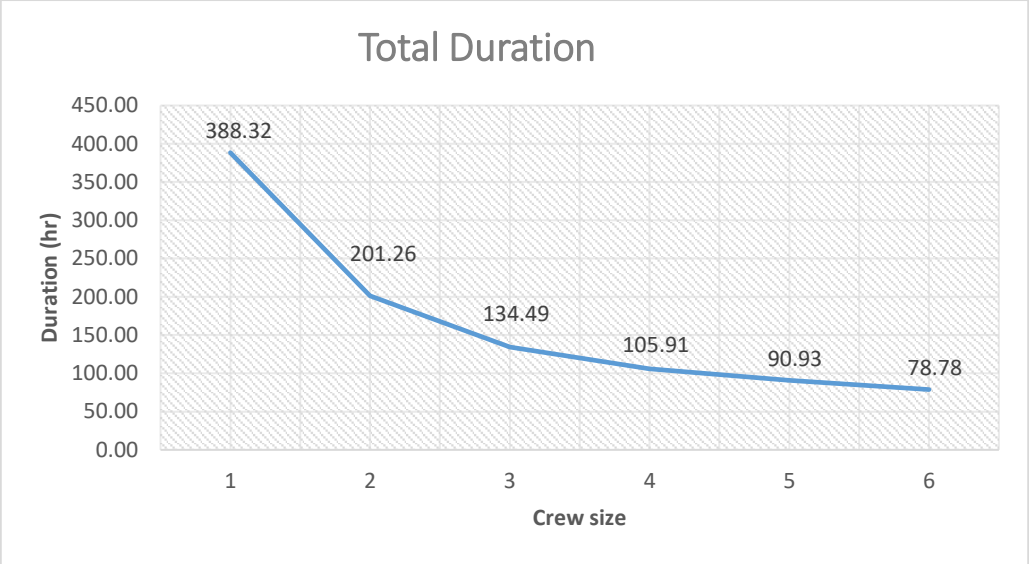


Figure 4.16. Task execution duration with different crew sizes with deterministic parameters

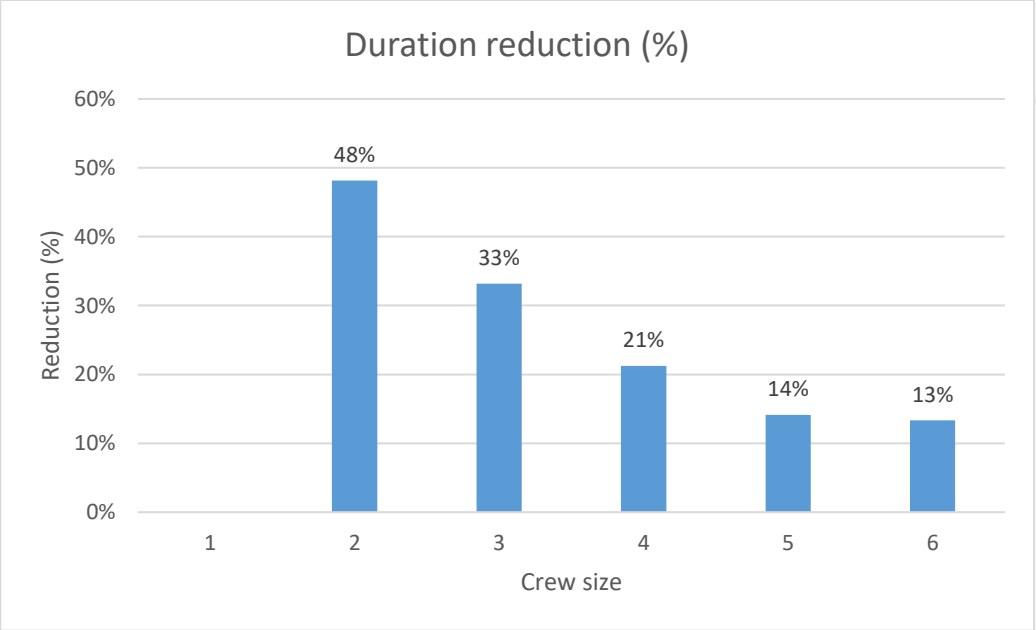


Figure 4.17. Task execution duration reduction by increasing crew size with deterministic parameters

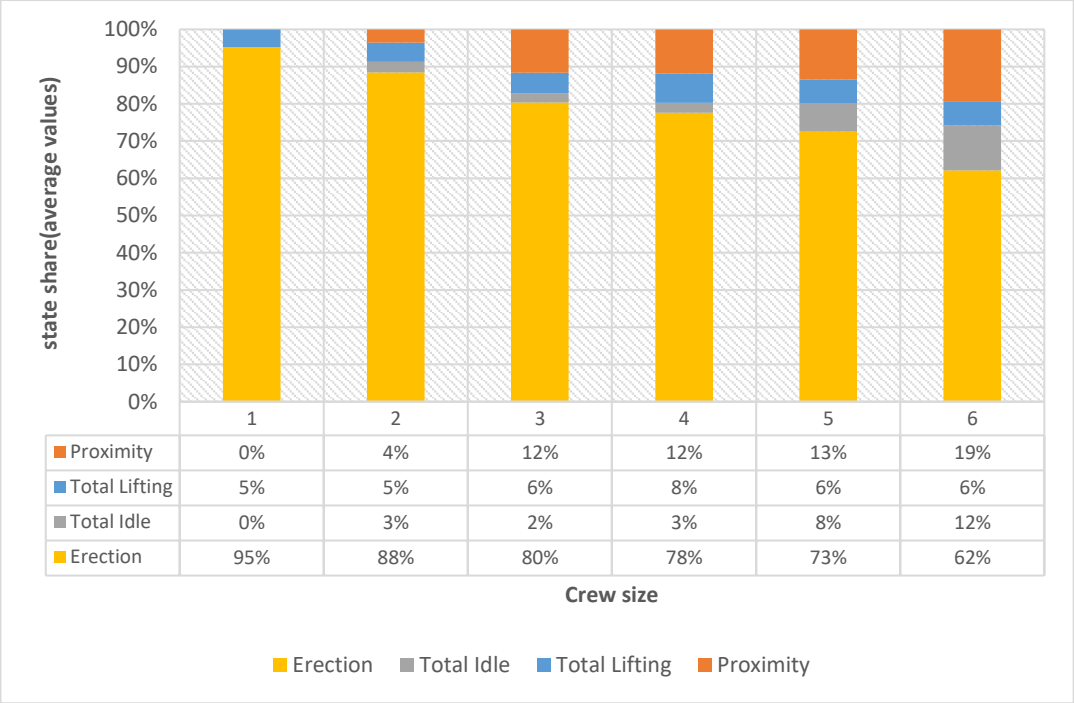


Figure 4.18. Stacked chart of the agent state during the simulation runs for the deterministic approach

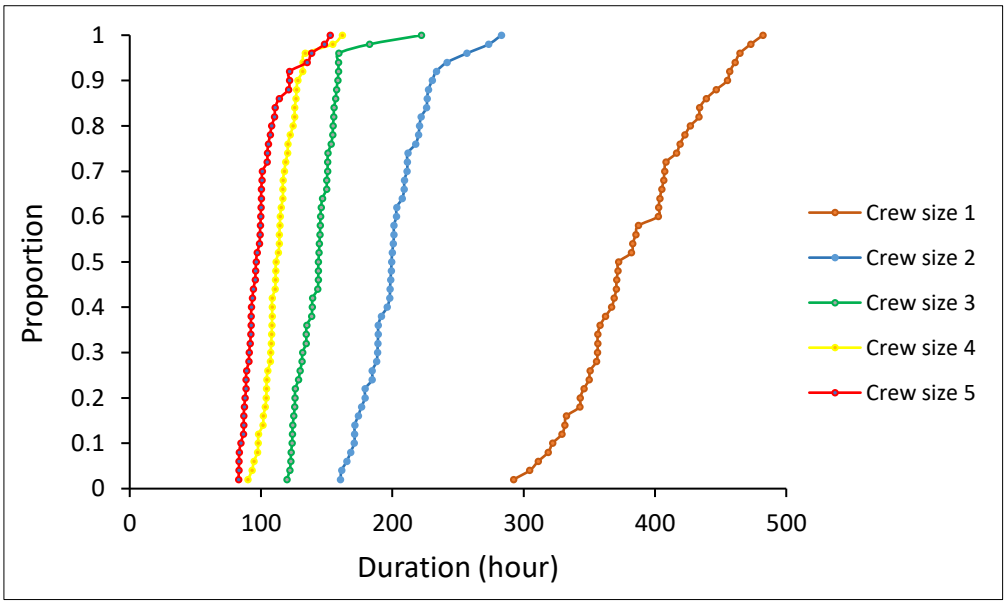


Figure 4.19. ECDF chart created by the simulation for different crew sizes

Table 4.1 Different duration percentiles and direct costs for different crew sizes

Crew Size	Duration of P10 (manhours)	Duration of P50 (manhours)	Duration of P90 (manhours)	Direct cost ($ per hour)
1	322	372	455	100
2	171	199	230	200
3	124	144	158	300
4	98	111	128	400
5	84	96	122	500

4.1.5 Discussion

The results demonstrated that time-space conflicts significantly impact crew performance, as the rate of duration reduction decreases with an increase in resources. Figure 4.20 shows the state proportion of an agent as the crew size increases from 1 to 5, based on the mean values of the results. The proximity state, which indicates time-space conflicts, operates independently of the other states. This provides users with sufficient data to make informed decisions tailored to the project's unique conditions using a quantitative approach.

To develop the project crashing modes, the direct cost of the corresponding resources is calculated based on the daily rental expenditure of mobile cranes per working hour (

Table 4.1), and the indirect cost of the associated activity is considered as $300 per working day. Herewith, a time-cost trade-off can be performed to achieve the best crashing scenario. Table 4.2 and Figure 4.21 shows different crashing modes for different crew sizes considering probability levels of %10, %50, and %90.

Mode-1 (Crew size =1) is dominated by Mode-2 (Crew size =2) as both the duration and cost values for all probability levels of Mode-2 are less than the corresponding values of Mode-1. Hence, Mode-1 leads to a non-optimal solution and should not be used unless there is a strict resource constraint. For the remaining modes, the crew size selection should be made considering the specific project and portfolio conditions. For example, for a contractor/subcontractor that does not want to complete the project early due to another project in which the contactor is planning to mobilize the crews and machinery after completion of this project, Mode-2 could be selected if the risk of exceeding the deadline for the project is not high (e.g. if the deadline is not much smaller than the 90% probability duration level of 230 hours). Mode-3 provides a balanced option for the time-cost trade-off and could be selected by a contractor that prefers the minimum budget risk as the 90% probability cost level of $53,325 is the lowest among the 90% probability cost level of all modes. Mode-4 and Mode-2 should be selected if the project has a tight deadline or if the contractor needs to mobilize the resources to another project earlier than the duration provided by Mode-3.

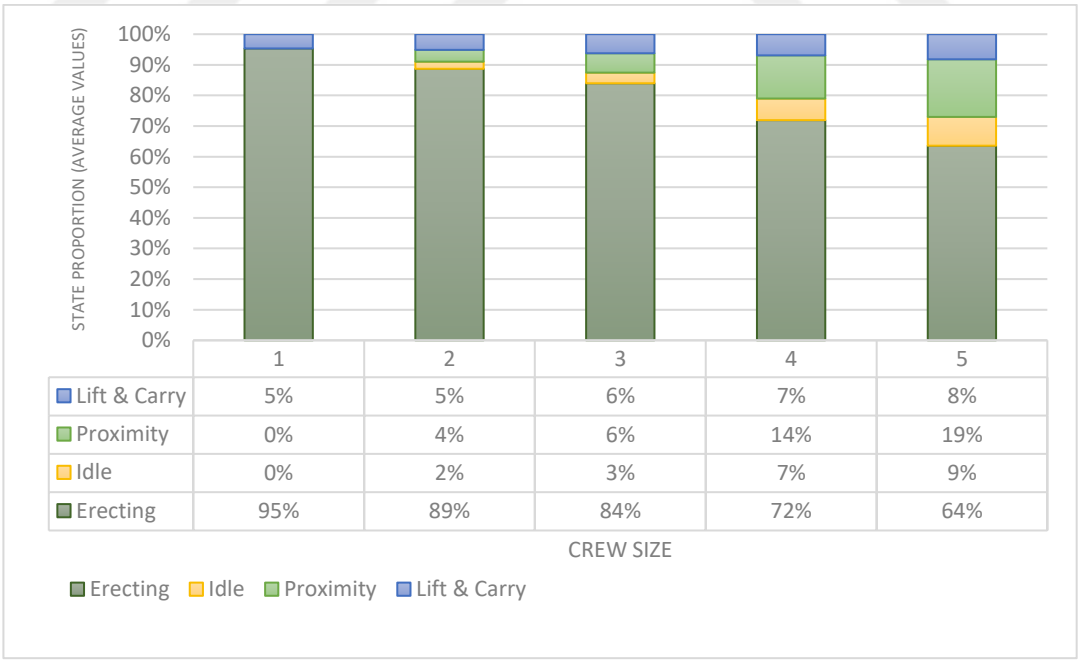


Figure 4.20. State proportion chart regarding an agent interaction in different simulation runs

Table 4.2 Crashing modes for various crew sizes

	Mode 1 [duration, cost] *	Mode 2 [duration, cost]	Mode 3 [duration, cost]	Mode 4 [duration, cost]	Mode 5 [duration, cost]
%10 percentile	[322, 44,275]	[171, 40,613]	[127, 41,850]	[98, 42,875]	[84, 45,150]
%50 percentile	[372, 51,150]	[199, 47,263]	[144, 48,600]	[111, 48,563]	[96, 51,600]
%90 percentile	[455, 62,563]	[230, 54,625]	[158, 53,325]	[128, 56,000]	[108, 58,050]

**duration: [hours]; cost: [$]*

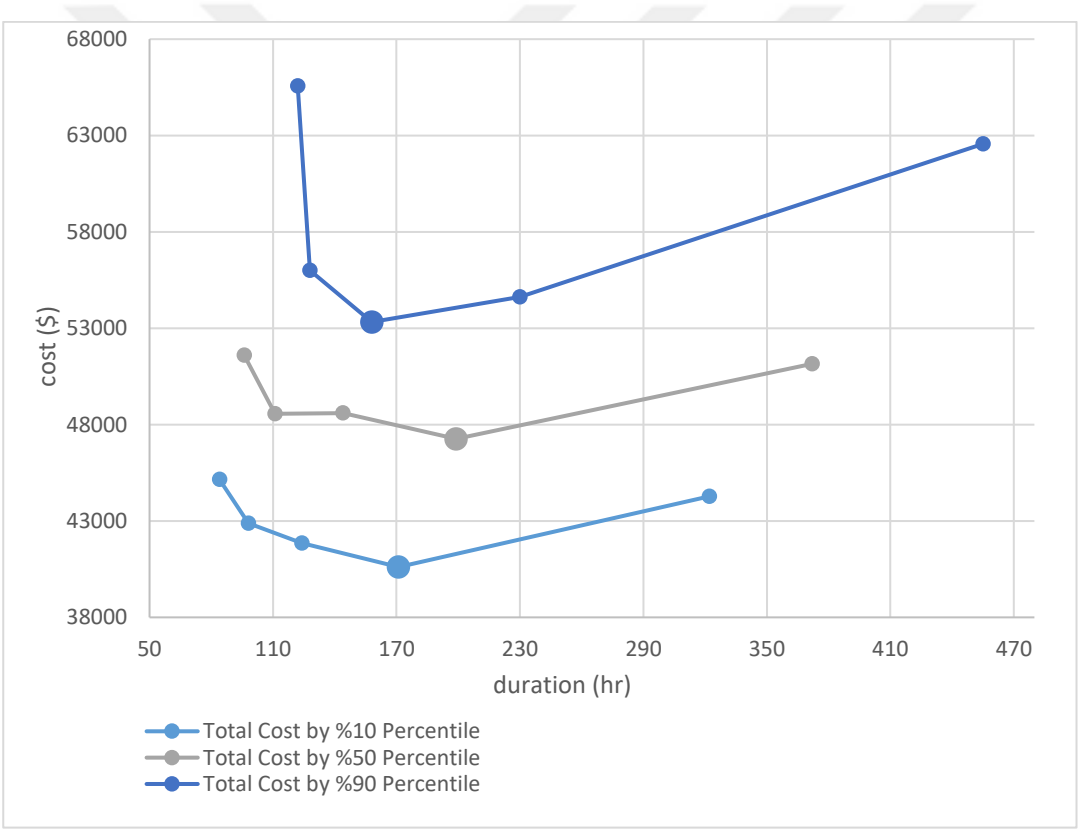


Figure 4.21. Crashing analysis to select the best crashing mode

4.1.6 Approach Limitations

Besides the considerable advantages gained by using the proposed game engine-based method, it has a few challenges. First, it includes complicated details like a 3D environment, independent agents, and dynamic space analyses. Because of this, the simulation duration is dominated by both software and hardware capabilities. In other words, increasing the hardware capabilities alone does not necessarily mean improving the simulation duration as the software limitations could be determinant in such cases. This means there's a cap on how fast it can run. If the run exceeds this limit, the system might start making mistakes because the trigger and sensory systems could fail. The simulation runs vary from 18 to 25 minutes depending on the crews' size and project conditions for each crashing alternative. This would make the simulations take much longer than usual operational or 4D simulations. In this regard, the system configurations for the simulation runs are provided in Table 4.3.

Table 4.3 The system configuration for simulation runs

Item	*Configuration*
Processor	12th Gen Intel(R) Core (TM) i7-12700H, 2300 Mhz, 14 Core(s), 20 Logical Processor(s)
RAM	32.0 GB
GPU	NVIDIA GeForce RTX 3050 GPU
Storage	1 TB SSD
System Type	x64-based PC
OS	Windows 11-Pro

4.2 Method Implementation in a Commercial Tool (Anylogic)

Anylogic is a commercial simulation tool supporting DES, ABS, and SD simulation types in the same platform. This feature makes possible the modeling of complex

systems with multiple distinct components. The heterogeneity model of the components can be achieved by combining different simulation types in the same model. Anylogic simulation engine is developed on Java programming language and can be packaged as a complete standalone Java application. Being tailored to Java, there are ready-to-use functions such as mathematics, probability distributions, time, and database functions, which facilitate efficient and streamlined development with almost no need for custom codes.

As discussed in section 2.3, Anylogic is not a sophisticated tool for direct integration of the BIM model and simulation environment, thus, the incorporation of the BIM model with the Anylogic can only be possible through a 2D plan or sectional views. Therefore, the visual fidelity of the developed simulation in this respect is considerably lower than the previous approach. On the other hand, this drawback brings the advantage of involving more concurrent agents in the simulation environment without much concern about the software or hardware incapability. This pair of limitations and benefits raises the need for a tradeoff between cost and benefit regarding the way of method implementation.

While the developed method relies on the ABS principles, the simulation benefits from other paradigm blocks to gather and analyze the simulation results. For instance, to calculate the revised productivity in a dynamic way the SD components of flow and stock are utilized.

4.2.1 Data Collection

The simulation relies on the data comprised of model data obtained from the BIM model through Daynomo and the project baseline schedule plan that is combined in a built-in database in Anylogic (Figure 4.22). The database is serverless and integrated with the model, meaning that it is cross-platform and transferable.

The data obtained from the baseline schedule plan consists of the assigned resources to the activity, their productivity, the criticality of the task, relationships, and the

planned start and finish dates. On the other hand, the model element data including the element ID, amount, and location is provided by the BIM model.

	id	activity	amount	unit	location	points	agent	size	productivity_per_working_hour	relationship	critical	element_id	start	finish
1	1	Earthwork	220	m3	foundation	1	excavator	5	3	-	1	303,540	29-02-2024...	08-03-2024...
2	2	Foundation- Reinforcement	16,236	kg	foundation	10	concreteCrew	5	20	F.S - 2	1	303,540	09-03-2024...	20-03-2024...
3	3	Foundation- Formwork	98.6	m	foundation	8	concreteCrew	3	5	F.S - 3	1	303,540	09-03-2024...	20-03-2024...
4	4	Foundation- Concrete	220	m3	foundation	2	concreteCrew	3	30	F.S - 4	1	303,540	21-03-2024...	21-03-2024...
5	5	Column - C-1-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4	1	303,798	22-03-2024...	22-03-2024...
6	6	Column - C-2-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4	1	303,948	22-03-2024...	22-03-2024...
7	7	Column - C-3-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4	1	303,953	22-03-2024...	22-03-2024...
8	8	Column - C-4-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4	1	303,958	22-03-2024...	22-03-2024...
9	9	Column - C-5-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9	1	303,963	23-03-2024...	23-03-2024...
10	10	Column - C-1-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9	1	303,944	23-03-2024...	23-03-2024...
11	11	Column - C-2-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9	1	303,949	23-03-2024...	23-03-2024...
12	12	Column - C-3-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9	1	303,954	23-03-2024...	23-03-2024...
13	13	Column - C-4-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12	1	303,959	24-03-2024...	24-03-2024...
14	14	Column - C-5-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12	1	303,964	24-03-2024...	24-03-2024...
15	15	Column - C-1-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12	1	303,945	24-03-2024...	24-03-2024...
16	16	Column - C-2-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12	1	303,950	24-03-2024...	24-03-2024...
17	17	Column - C-3-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16	1	303,955	25-03-2024...	25-03-2024...
18	18	Column - C-4-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16	1	303,960	25-03-2024...	25-03-2024...
19	19	Column - C-5-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16	1	303,965	25-03-2024...	25-03-2024...
20	20	Column - C-1-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16	1	303,946	25-03-2024...	25-03-2024...
21	21	Column - C-2-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20	1	303,951	26-03-2024...	26-03-2024...
22	22	Column - C-3-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20	1	303,956	26-03-2024...	26-03-2024...
23	23	Column - C-4-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20	1	303,961	26-03-2024...	26-03-2024...
24	24	Column - C-5-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20	1	303,966	26-03-2024...	26-03-2024...
25	25	Column - C-1-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24	1	303,947	27-03-2024...	27-03-2024...
26	26	Column - C-2-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24	1	303,952	27-03-2024...	27-03-2024...
27	27	Column - C-3-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24	1	303,957	27-03-2024...	27-03-2024...
28	28	Column - C-4-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24	1	303,962	27-03-2024...	27-03-2024...
29	29	Column - C-5-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 28	1	303,967	28-03-2024...	28-03-2024...
30	30	Columns Formwork	160	m2	level 1	15	concreteCrew	4	10	F.S - 29	1	0	28-03-2024...	30-03-2024...
31	31	Columns Concrete	12	m3	level 1	15	concreteCrew	4	1.5	F.S - 30	1	0	31-03-2024...	31-03-2024...
32	32	Beams & Floor Formwork	758.27	m2	level 2	40	concreteCrew	4	10	F.S - 31	1	0	01-04-2024...	11-04-2024...

Figure 4.22. The built-in database in Anylogic with BIM model data and baseline schedule

4.2.2 Agent Statechart

Each agent holds a unique set of behaviors defined by a chart called statechart. Statechart represents the time- or event-driven operations in a sequential practice based on the agent's perception of the surrounding environment and consists of transitions and states with an entry point. Transitions are prerequisites for changes in the state of agents, which can be triggered by simple timeouts or more complex initiates such as receiving a message resulting from interactions between agents. Triggering a transition assigns a new state to the agent that can be followed by the execution of specific reactions. States are the control point of agent behavior concerning the mentioned interactions. A state can be a simple or a combination of states called a composite state. The composite state has an entry point that represents its starting state, meaning that whenever the composite state is entered, the entry

point redirects to the associated entry state. Nevertheless, this behavior may not always be needed, as there can be situations where the last taken state within the composite state should be remembered. In such cases, when the composite state is re-entered, the statechart should return to the last active state rather than starting from the entry point. In such cases, the history state should be added to the composite state. As the name suggests, history states hold the last visited state within a composite state as shown in Figure 4.24. Some of the agents used in the developed simulation are listed in the following subsections.

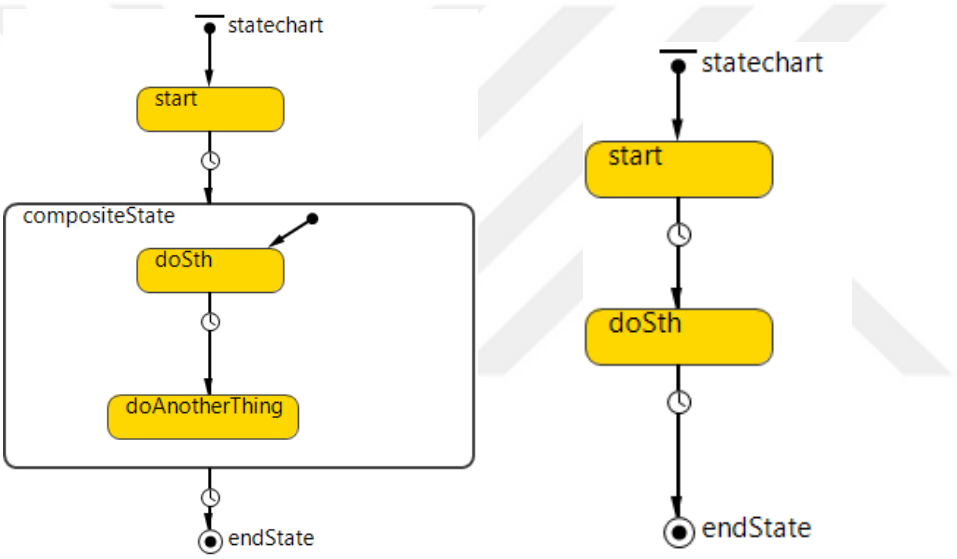


Figure 4.23. Simple state (right) and composite state (left)

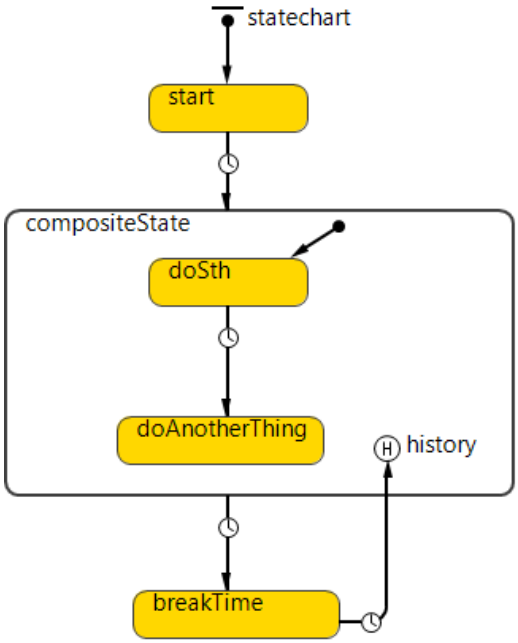


Figure 4.24. Using history state to remember the last visited state

4.2.2.1 Worker agent

This agent is a superclass representing all the crew agents in the simulation and all other crew agents are a subclass of the worker agent. This relationship is called “inheritance” in Java in which the subclasses are derived from a superclass inheriting its methods and fields. Figure 4.25 indicates an example of inheritance in Java where student, engineer, and writer classes are subclasses of the human class. Inheritance is useful to avoid unnecessary detailing and provide reusability of code. In this context, the worker agent provides the fields and functions that are common throughout the other crew agents. The fields and functions such as calculating the *r* factor, conflict detection, and reporting are inherited from the worker class. Figure 4.26 depicts the environment of the worker agent.

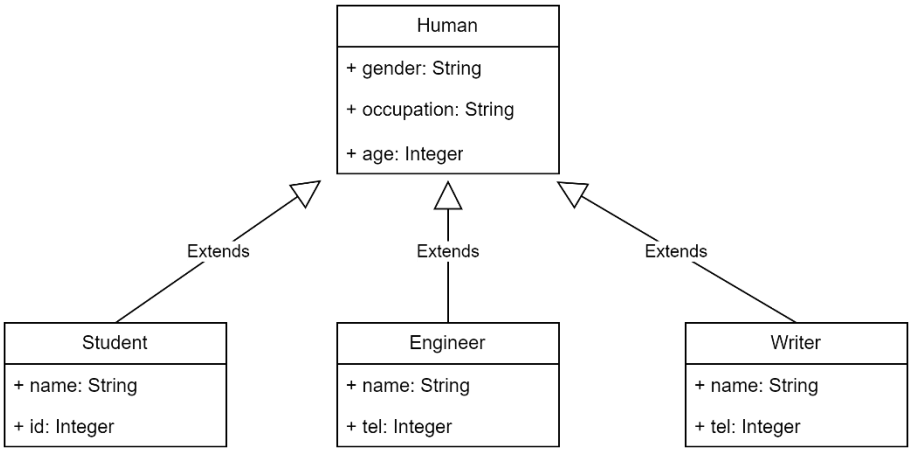


Figure 4.25. Inheritance in Java programming language

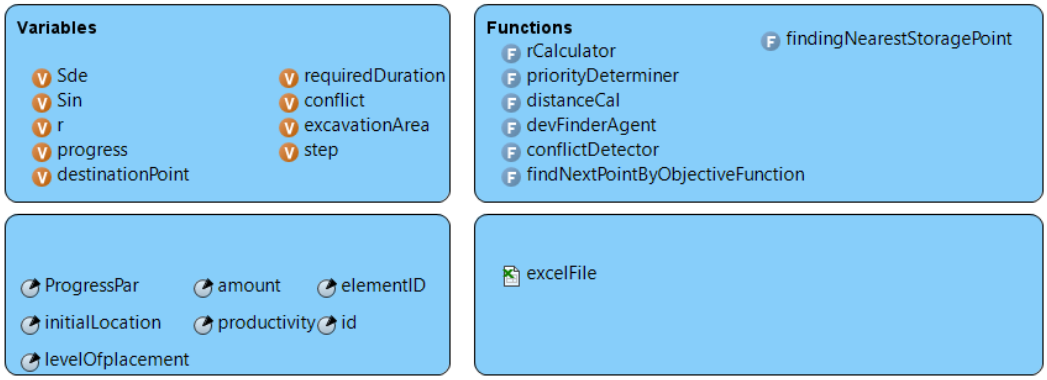


Figure 4.26. The worker agent's environment

4.2.2.2 Excavator agent

In the current study, each agent owns a unique statechart representing its characteristics and reactions to the simulation events. For instance, the statechart of the excavation crew includes idle, surveying, excavation, transporting soil to off-site, and grading states, which is presented in Figure 4.27.

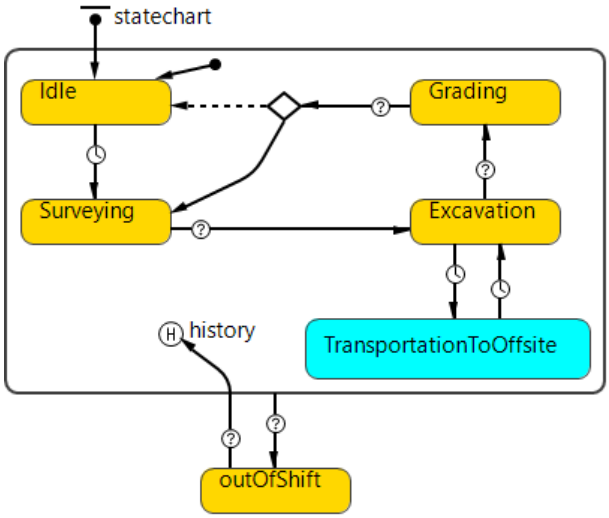


Figure 4.27. The excavator agent environment with statechart

There is also a composite state responsible for developing out-of-shift behavior for non-working hours and weekdays. A history state is used to remember the last visited state by the agent so that, upon returning to in-shift working hours, the agent can continue from where they left off. Both timeout (clock icon) and conditional transitions (question mark icon) are used to mimic the crew's behavior regarding the interaction with other agents or the environment. Each transition triggers the embedded scripts alongside the state change.

By adding any agents to the simulation, the idle state is taken initially. Entering this state triggers a set of state scripts accordingly. Figure 4.28 and Figure 4.29 demonstrate the scripts run by an on-enter trigger of idle state for the excavation agent and the agent environment with the associated fields for the excavator agent respectively. The fields ending with the “State” suffix are used to collect the time spent in each state. These fields are the stock blocks borrowed from the SD simulation type. The stocks are filled according to a dynamic check mechanism where the flow to the stocks is triggered according to the state change. Apart from the state change, the conflict also triggers a flow to the conflict stock block, which is used to track conflict cases. The conflict is determined using a function called *rCalculator* that is used to calculate the *r* factor. The associated code for the function

is given in Figure 4.30 with an assumed cubic workspace of a cube with 2 meters on all edges.

```
SurveyingState = 0;
ExcavationState = 0;
GradingState = 0;
progress =0;
selectedNodeFromAreaComponent = main.foundationLevelZones.get(uniform_discr(0,
main.foundationLevelZones.size()-1));
main.foundationLevelZones.remove(selectedNodeFromAreaComponent);
```

Figure 4.28. The script triggered by taking the idle state

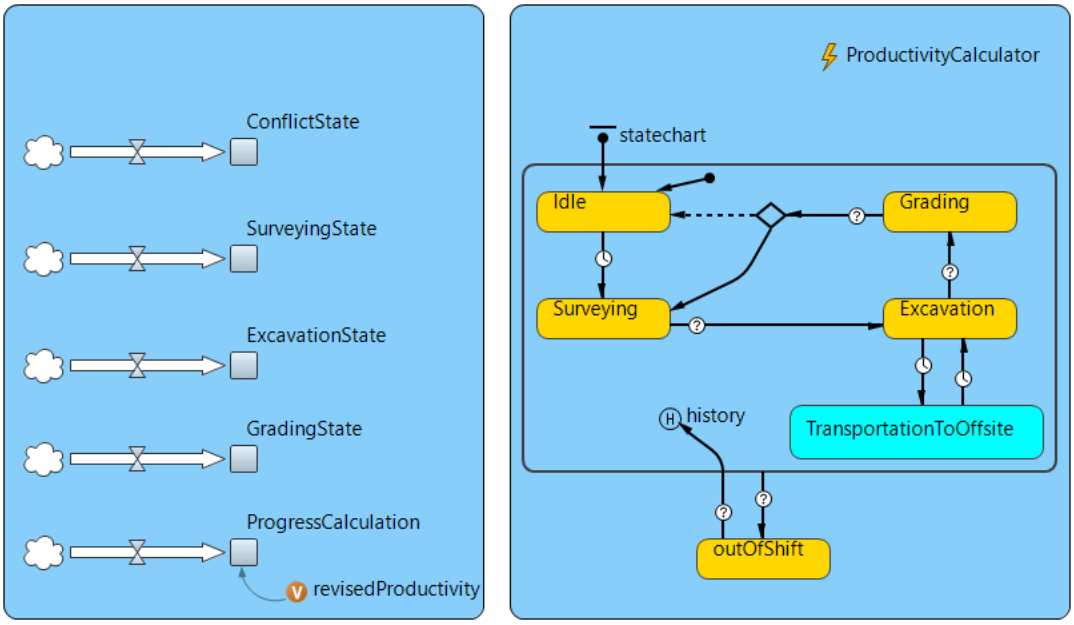


Figure 4.29. The agent environment for the excavator agent

Any conflict within the agents that does not lead to constructability issue, updates the baseline productivity based on the *r* factor. The updated productivity is assigned to a variable called *revisedProductivity* which is used to calculate the progress of the task for the current agent.

Following the idle state, whenever the task is initiated, a time-out transition, which represents the time needed to reach the target level, is triggered which takes the agent to the next state of surveying. Taking the surveying state, the agent moves to the destination point by a variable called *destinationPoint* inherited from the worker

class. This point is selected by considering the objective function mentioned in Chapter 3 and the relevant codes are given in Figure 4.31.

```
r =1;
Sde = getLength() * getWidth() * getHeight();
Sin =0;
double XminA=getPosition().x -getLength()/2 ;
double XmaxA=getPosition().x +getLength()/2 ;
double YminA=getPosition().y -getWidth()/2;
double YmaxA=getPosition().y +getWidth()/2;
double ZminA=getPosition().z -getHeight()/2;
double ZmaxA=getPosition().z +getHeight()/2;
double XminB = 0; double XmaxB = 0; double YminB = 0; double YmaxB = 0;
double ZminB = 0;
double ZmaxB = 0;
double x =0; double y=0; double z=0;

if (agentsInRange(2).size()>0) {
 XminB=agentsInRange(2).get(0).getPosition().x -
(agentsInRange(2).get(0).getLength())/2;
 XmaxB=agentsInRange(2).get(0).getPosition().x
+(agentsInRange(2).get(0).getLength())/2;
 YminB=agentsInRange(2).get(0).getPosition().y -
(agentsInRange(2).get(0).getWidth())/2;
 YmaxB=agentsInRange(2).get(0).getPosition().y
+(agentsInRange(2).get(0).getWidth())/2;
 ZminB=agentsInRange(2).get(0).getPosition().z -
(agentsInRange(2).get(0).getHeight())/2;
 ZmaxB=agentsInRange(2).get(0).getPosition().z
+(agentsInRange(2).get(0).getHeight())/2;

 x =abs( min(XmaxA, XmaxB) - max(XminB, XminA));
 y =abs( min(YmaxA,YmaxB) - max(YminB, YminA));
 z =abs( min(ZmaxA,ZmaxB) - max(ZminB, ZminA));
 Sin = x*y*z;
}
else Sin = 0;

 if (Sin>0){
 conflict = true;}
 else conflict = false;

 r=1-(Sin/Sde);

return r;
```

Figure 4.30. Calculating *r* factor using Java code in Anylogic

4.2.2.3 ConcreteCrew agent

This agent represents the crew for concrete work including formwork, rebar, and concrete tasks. As indicated before, the agent inherits the fields and functions of the worker class, however, there is a unique statechart representing its distinct behaviors (Figure 4.32). There is a need to visit the supply point to collect the materials in this

agent, therefore, the state of “supply” is provided. The user determines the supply point in the simulation startup configuration, or it can be calculated by mathematical and geometric approaches such as the centroid of the plan shape.

```
double sum=0;
double ave=0;
double dev=0;
ArrayList<Double> d = new ArrayList<Double>();
ArrayList<PointNode> d2 = new ArrayList<PointNode>();
Hashtable<Double, PointNode> h = new Hashtable<Double,
PointNode>();
if (otherAgentPositions.size()==1) {
for (PointNode c1: potentialPoints){
if (sum < distanceCal(c1, otherAgentPositions.get(0))){
sum = distanceCal(c1, otherAgentPositions.get(0));
h.put(sum, c1);
}
}
destinationPoint= h.get(sum);
h.clear();
}

if (otherAgentPositions.size()>1){
for (PointNode c1: potentialPoints){
for (PointNode c2: otherAgentPositions){
sum = distanceCal(c1, c2)+sum;
}
ave= sum/otherAgentPositions.size();
for (PointNode c2: otherAgentPositions){
d.add(abs(distanceCal(c2, c1)-ave));
}
Collections.sort(d, Comparator.reverseOrder());
dev=d.get(0);
h.put(dev, c1);
}
Iterator it = h.keys().asIterator();
dev = 99999;
while (it.hasNext()){
double s = Double.parseDouble(it.next().toString());
if (s < dev){
dev = s;}
}
destinationPoint = h.get(dev);
h.clear();
}
```

Figure 4.31. Finding destination point in Java code

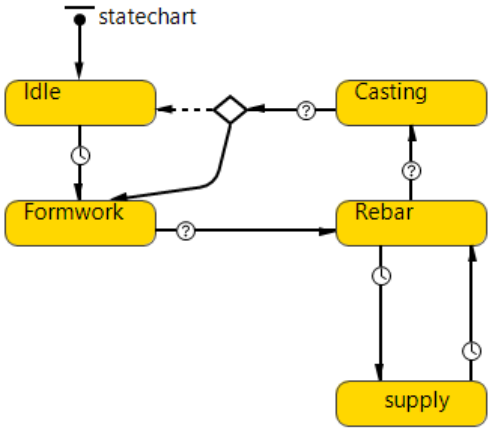


Figure 4.32. The statechart of the concrete crew agent

The agent should visit the supply point regularly to collect and bend the required rebars. This process is triggered by a timeout transition with a probabilistic value determined by the user.

4.2.2.4 Masonry agent

This agent represents the masonry crew for executing different types of walls. The supply state visit happens in two states of mortar creation and placing bricks and blocks.

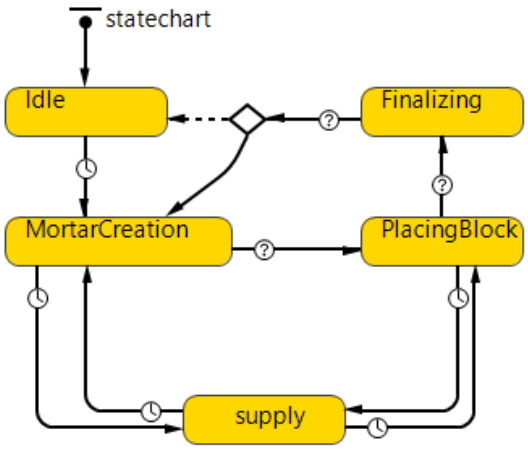


Figure 4.33. The masonry agent statechart

4.2.3 The Simulation Environment

The simulation environment is created in 2D using plan views of the BIM model (see Figure 4.34). The views are taken from various levels and the model elements are placed on the plan using Anylogic built-in space markups. The element IDs are used to map the database entities with the environment elements. Each agent moves in the environment considering the associated statechart and obstacles of stationary and moving objects. Any under-execution or installed objects becomes an obstacle so the rest of the agents avoid them accordingly.

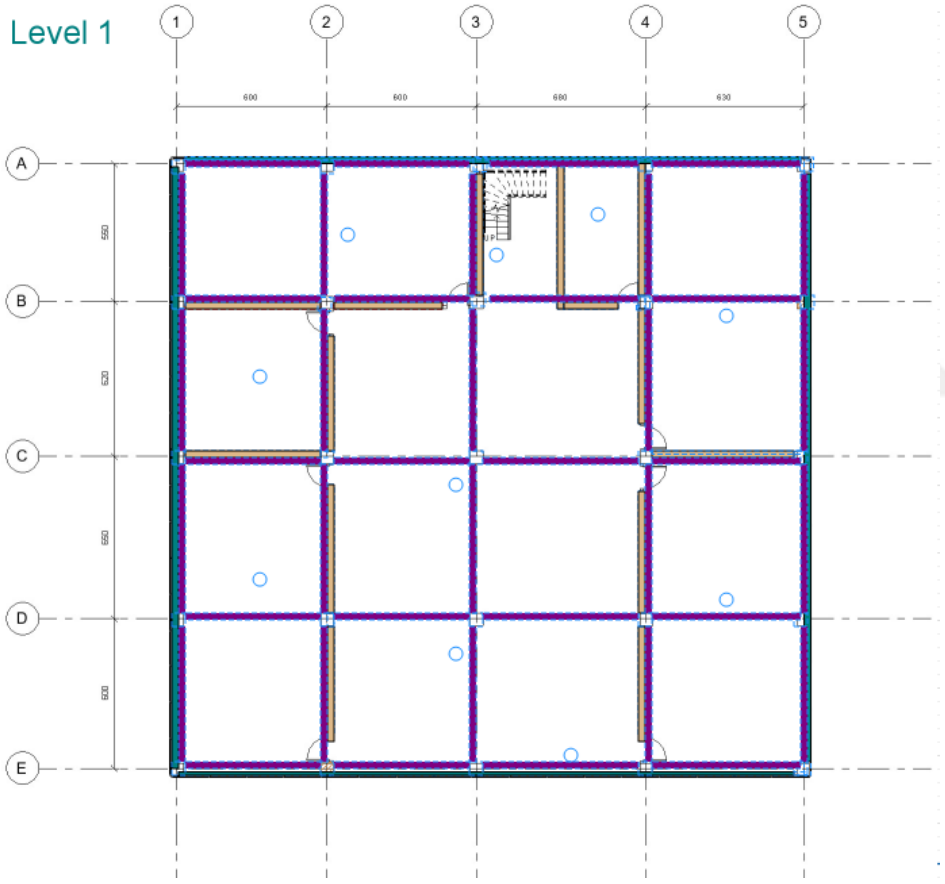


Figure 4.34. BIM model plan view as the simulation environment

A scheduling script is used to fetch daily planned tasks through a Java collection type called tuple from the database and assign them to the associated agents. Furthermore, it monitors the working days and hours and enables out-of-shift work pauses. If there is no available agent to accept the daily planned tasks, due to the previously delayed tasks, the planned task is submitted to the waiting task list to be handled once the agent is available. Accordingly, the actual start and finish dates of a task are added to the database for associated tasks.

```
if ( dailyTaskDeterminer.getValue()) {
try {
Date d = date();
todayTasks = new ArrayList(workplanMap
.get(new Date(d.getYear(), d.getMonth(), d.getDate())));
for (Tuple t: todayTasks) {
String associatedLevel = t.get(11, String.class);
String agentName = t.get(3, String.class);
int size = t.get(15, Integer.class);
int id = t.get(6, Integer.class);
int elementId = t.get(7,Integer.class);
double amount = t.get(5,Double.class);
double productivity = t.get(13,Double.class);
for (int i=0; i<size; ++i) {
addAgentToSimulationFunc(agentName, productivity/size, elementId, amount/size,
size,id);
}
}
}
catch (Exception e) {
traceln("there is no workplan for "+ time());
}
}
```

Figure 4.35. The schedule block for daily work assignments

4.2.4 The Simulation Run

The simulation is run with the associated parameters on the project model indicated in Figure 4.36. The associated parameter values are taken from the RSMeans databases and engineering judgments wherever required. The time-space conflicts are detected based on the agents' movement and reflected in the agent's productivity or workability. Consequently, the baseline schedule is modified with simulation-based start and finish dates of the project activities (Figure 4.38). These values are

recorded on the database for further analysis of crash modes (Figure 4.37). The revised dates accompanied by the cost of resource increase enable the time-cost trade-off over the modes to achieve the best scenario fit for the project conditions. The model time steps are in discrete minutes while the analyzing blocks are in continuous time steps according to SD principles.

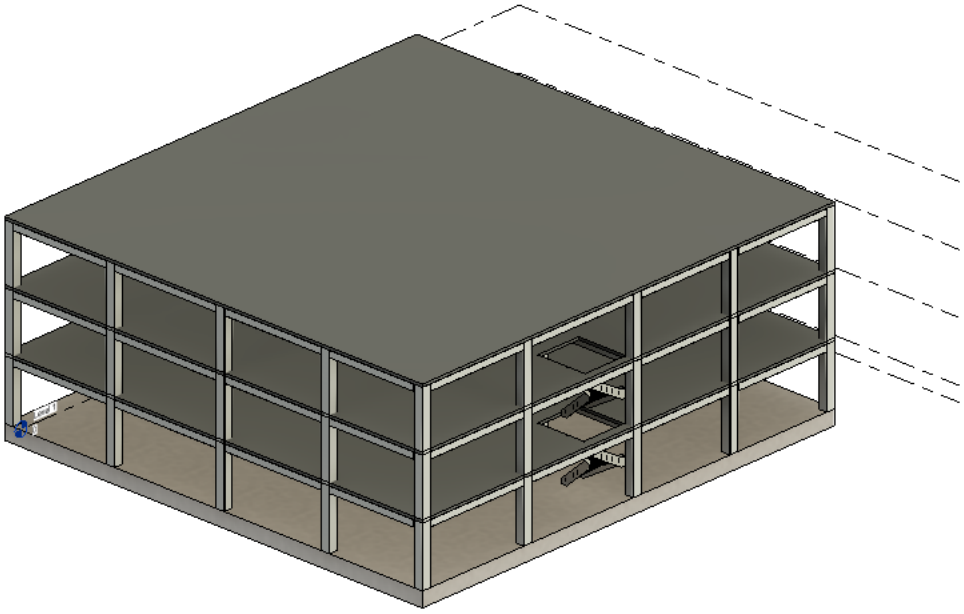


Figure 4.36. The case 2 project model in Autodesk Revit

The list of the activities and the baseline schedule plan Gantt chart are presented in Figure 4.41 and Figure 4.40. It is assumed that the project undergoes a delay and there is a need to expedite the activities on the critical path by injecting more resources (crashing). The project start time is set to 29/02/2024 with a total duration of 91 days for structural elements. The crashing process starts from task ID 20 by allocating more concrete crew resources in two modes. The resource allocation in the baseline schedule is a total of 1056 manhours for concrete works including rebar, formwork, and concrete casting, and 72 manhours for the excavation works. Increasing the resource allocation incurs higher costs that should be appraised

alongside the saved project's overall duration to end up with an acceptable time and cost for the project.

Main | workplan

	agent	size	productivity_per_working_hour	relationship	critical	element_id	start	finish	actual_start	actual_finsish
1	excavator	5	3	-	1	303,540	29-02-2024 17:...	08-03-2024 17:...	29-02-2024 08:...	13-03-2024 15:...
2	concreteCrew	5	20	F.S - 2	1	303,540	09-03-2024 18:...	20-03-2024 18:...	14-03-2024 15:...	27-03-2024 18:...
3	concreteCrew	3	5	F.S - 3	1	303,540	09-03-2024 18:...	20-03-2024 18:...	14-03-2024 15:...	27-03-2024 18:...
4	concreteCrew	3	30	F.S - 4	1	303,540	21-03-2024 18:...	21-03-2024 18:...	28-03-2024 18:...	29-03-2024 18:...
5	concreteCrew	2	0.2	F.S - 4	1	303,798	22-03-2024 18:...	22-03-2024 18:...	30-03-2024 18:...	31-03-2024 18:...
6	concreteCrew	2	0.2	F.S - 4	1	303,948	22-03-2024 18:...	22-03-2024 18:...	30-03-2024 18:...	31-03-2024 18:...
7	concreteCrew	2	0.2	F.S - 4	1	303,953	22-03-2024 18:...	22-03-2024 18:...	30-03-2024 18:...	31-03-2024 18:...
8	concreteCrew	2	0.2	F.S - 4	1	303,958	22-03-2024 18:...	22-03-2024 18:...	30-03-2024 18:...	31-03-2024 18:...
9	concreteCrew	2	0.2	F.S - 9	1	303,963	23-03-2024 18:...	23-03-2024 18:...	01-04-2024 18:...	01-04-2024 18:...
10	concreteCrew	2	0.2	F.S - 9	1	303,944	23-03-2024 18:...	23-03-2024 18:...	01-04-2024 18:...	01-04-2024 18:...
11	concreteCrew	2	0.2	F.S - 9	1	303,949	23-03-2024 18:...	23-03-2024 18:...	01-04-2024 18:...	01-04-2024 18:...
12	concreteCrew	2	0.2	F.S - 9	1	303,954	23-03-2024 18:...	23-03-2024 18:...	01-04-2024 18:...	01-04-2024 18:...
13	concreteCrew	2	0.2	F.S - 12	1	303,959	24-03-2024 18:...	24-03-2024 18:...	02-04-2024 18:...	02-04-2024 18:...
14	concreteCrew	2	0.2	F.S - 12	1	303,964	24-03-2024 18:...	24-03-2024 18:...	02-04-2024 18:...	02-04-2024 18:...
15	concreteCrew	2	0.2	F.S - 12	1	303,945	24-03-2024 18:...	24-03-2024 18:...	02-04-2024 18:...	03-04-2024 18:...
16	concreteCrew	2	0.2	F.S - 12	1	303,950	24-03-2024 18:...	24-03-2024 18:...	03-04-2024 18:...	03-04-2024 18:...
17	concreteCrew	2	0.2	F.S - 16	1	303,955	25-03-2024 18:...	25-03-2024 18:...	05-04-2024 18:...	05-04-2024 18:...
18	concreteCrew	2	0.2	F.S - 16	1	303,960	25-03-2024 18:...	25-03-2024 18:...	05-04-2024 18:...	05-04-2024 18:...
19	concreteCrew	2	0.2	F.S - 16	1	303,965	25-03-2024 18:...	25-03-2024 18:...	05-04-2024 18:...	06-04-2024 18:...
20	concreteCrew	2	0.2	F.S - 16	1	303,946	25-03-2024 18:...	25-03-2024 18:...	06-04-2024 18:...	07-04-2024 18:...
21	concreteCrew	2	0.2	F.S - 20	1	303,951	26-03-2024 18:...	26-03-2024 18:...	07-04-2024 18:...	07-04-2024 18:...
22	concreteCrew	2	0.2	F.S - 20	1	303,956	26-03-2024 18:...	26-03-2024 18:...	07-04-2024 18:...	08-04-2024 18:...
23	concreteCrew	2	0.2	F.S - 20	1	303,961	26-03-2024 18:...	26-03-2024 18:...	08-04-2024 18:...	08-04-2024 18:...
24	concreteCrew	2	0.2	F.S - 20	1	303,966	26-03-2024 18:...	26-03-2024 18:...	08-04-2024 18:...	09-04-2024 18:...
25	concreteCrew	2	0.2	F.S - 24	1	303,947	27-03-2024 18:...	27-03-2024 18:...	09-04-2024 18:...	09-04-2024 18:...
26	concreteCrew	2	0.2	F.S - 24	1	303,952	27-03-2024 18:...	27-03-2024 18:...	09-04-2024 18:...	10-04-2024 18:...
27	concreteCrew	2	0.2	F.S - 24	1	303,957	27-03-2024 18:...	27-03-2024 18:...	10-04-2024 18:...	10-04-2024 18:...
28	concreteCrew	2	0.2	F.S - 24	1	303,962	27-03-2024 18:...	27-03-2024 18:...	10-04-2024 18:...	11-04-2024 18:...
29	concreteCrew	2	0.2	F.S - 28	1	303,967	28-03-2024 18:...	28-03-2024 18:...	11-04-2024 18:...	11-04-2024 18:...
30	concreteCrew	4	10	F.S - 29	1	0	28-03-2024 18:...	30-03-2024 18:...	11-04-2024 18:...	16-04-2024 18:...
31	concreteCrew	4	1.5	F.S - 30	1	0	31-03-2024 18:...	31-03-2024 18:...	17-04-2024 18:...	17-04-2024 18:...
32	concreteCrew	4	10	F.S - 31	1	0	01-04-2024 18:...	11-04-2024 18:...	18-04-2024 18:...	01-05-2024 18:...

Figure 4.37. The revised schedule with simulation-based start and finish dates

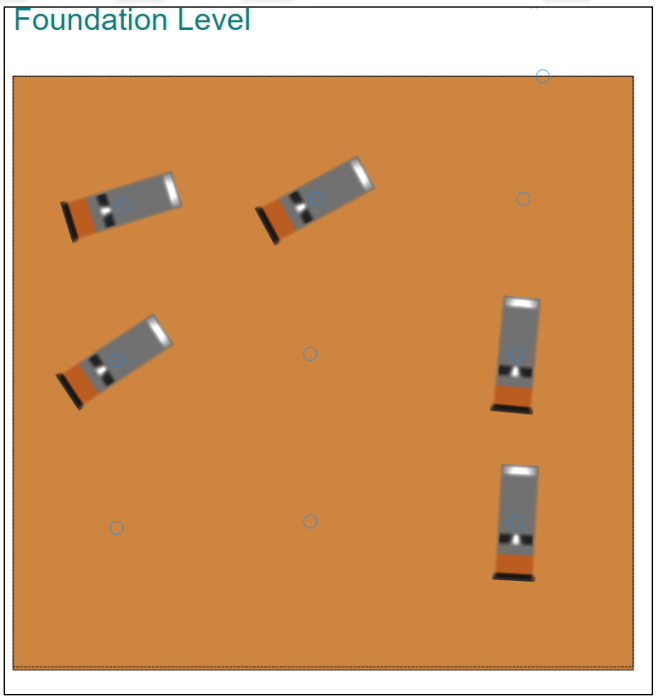


Figure 4.38. A simulation run scene in Anylogic

ID	Name	Work Amount	Unit	Productivity (work amount one crew per day)	Crew size	Start	Finish	Duration
1	Earthwork	220	m3	24.4	1	2/29/2024	3/8/2024	9
2	Foundation- Reinforcement	16236	kg	135.3	10	3/9/2024	3/20/2024	12
3	Foundation- Formwork	98.6	m	1.0	8	3/9/2024	3/20/2024	12
4	Foundation- Concrete	220	m3	110.0	2	3/21/2024	3/21/2024	1
5	Column - C-1-A	0.8	m3	0.8	1	3/22/2024	3/22/2024	1
6	Column - C-2-A	0.8	m3	0.8	1	3/22/2024	3/22/2024	1
7	Column - C-3-A	0.8	m3	0.8	1	3/22/2024	3/22/2024	1
8	Column - C-4-A	0.8	m3	0.8	1	3/22/2024	3/22/2024	1
9	Column - C-5-A	0.8	m3	0.8	1	3/23/2024	3/23/2024	1
10	Column - C-1-B	0.8	m3	0.8	1	3/23/2024	3/23/2024	1
11	Column - C-2-B	0.8	m3	0.8	1	3/23/2024	3/23/2024	1
12	Column - C-3-B	0.8	m3	0.8	1	3/23/2024	3/23/2024	1
13	Column - C-4-B	0.8	m3	0.8	1	3/24/2024	3/24/2024	1
14	Column - C-5-B	0.8	m3	0.8	1	3/24/2024	3/24/2024	1
15	Column - C-1-C	0.8	m3	0.8	1	3/25/2024	3/25/2024	1
16	Column - C-2-C	0.8	m3	0.8	1	3/24/2024	3/24/2024	1
17	Column - C-3-C	0.8	m3	0.8	1	3/25/2024	3/25/2024	1
18	Column - C-4-C	0.8	m3	0.8	1	3/25/2024	3/25/2024	1
19	Column - C-5-C	0.8	m3	0.8	1	3/25/2024	3/25/2024	1
20	Column - C-1-D	0.8	m3	0.8	1	3/25/2024	3/25/2024	1
21	Column - C-2-D	0.8	m3	0.8	1	3/26/2024	3/26/2024	1
22	Column - C-3-D	0.8	m3	0.8	1	3/26/2024	3/26/2024	1
23	Column - C-4-D	0.8	m3	0.8	1	3/27/2024	3/27/2024	1
24	Column - C-5-D	0.8	m3	0.8	1	3/27/2024	3/27/2024	1
25	Column - C-1-E	0.8	m3	0.8	1	3/28/2024	3/28/2024	1
26	Column - C-2-E	0.8	m3	0.8	1	3/28/2024	3/28/2024	1
27	Column - C-3-E	0.8	m3	0.8	1	3/29/2024	3/29/2024	1
28	Column - C-4-E	0.8	m3	0.8	1	3/29/2024	3/29/2024	1
29	Column - C-5-E	0.8	m3	0.8	1	3/30/2024	3/30/2024	1
30	Columns Formwork	160	m2	2.9	8	3/31/2024	4/6/2024	7
31	Columns Concrete	12	m3	0.5	8	4/5/2024	4/7/2024	3
32	Beams & Floor Formwork	758.27	m2	5.3	8	4/8/2024	4/25/2024	18
33	Beam - A-1-2	0.354	m3	0.4	1	4/26/2024	4/26/2024	1
34	Beam - A-2-3	0.354	m3	0.4	1	4/26/2024	4/26/2024	1
35	Beam - A-3-4	0.405	m3	0.4	1	4/27/2024	4/27/2024	1
36	Beam - A-4-5	0.372	m3	0.4	1	4/27/2024	4/27/2024	1
37	Beam - B-1-2	0.354	m3	0.4	1	4/28/2024	4/28/2024	1
38	Beam - B-2-3	0.354	m3	0.4	1	4/28/2024	4/28/2024	1
39	Beam - B-3-4	0.405	m3	0.4	1	4/29/2024	4/29/2024	1
40	Beam - B-4-5	0.372	m3	0.4	1	4/29/2024	4/29/2024	1
41	Beam - C-1-2	0.354	m3	0.4	1	4/30/2024	4/30/2024	1
42	Beam - C-2-3	0.354	m3	0.4	1	4/30/2024	4/30/2024	1
43	Beam - C-3-4	0.405	m3	0.4	1	5/1/2024	5/1/2024	1
44	Beam - C-4-5	0.372	m3	0.4	1	5/1/2024	5/1/2024	1
45	Beam - D-1-2	0.354	m3	0.4	1	5/2/2024	5/2/2024	1
46	Beam - D-2-3	0.354	m3	0.4	1	5/2/2024	5/2/2024	1
47	Beam - D-3-4	0.405	m3	0.4	1	5/3/2024	5/3/2024	1
48	Beam - D-4-5	0.372	m3	0.4	1	5/3/2024	5/3/2024	1
49	Beam - E-1-2	0.354	m3	0.4	1	5/4/2024	5/4/2024	1
50	Beam - E-2-3	0.354	m3	0.4	1	5/4/2024	5/4/2024	1
51	Beam - E-3-4	0.405	m3	0.4	1	5/5/2024	5/5/2024	1
52	Beam - E-4-5	0.372	m3	0.4	1	5/5/2024	5/5/2024	1
53	Beam - 1-A-B	0.321	m3	0.3	1	5/6/2024	5/6/2024	1
54	Beam - 2-A-B	0.321	m3	0.3	1	5/6/2024	5/6/2024	1
55	Beam - 3-A-B	0.321	m3	0.3	1	5/7/2024	5/7/2024	1
56	Beam - 4-A-B	0.321	m3	0.3	1	5/7/2024	5/7/2024	1
57	Beam - 5-A-B	0.321	m3	0.3	1	5/8/2024	5/8/2024	1
58	Beam - 1-B-C	0.367	m3	0.4	1	5/8/2024	5/8/2024	1
59	Beam - 2-B-C	0.367	m3	0.4	1	5/9/2024	5/9/2024	1
60	Beam - 3-B-C	0.367	m3	0.4	1	5/9/2024	5/9/2024	1
61	Beam - 4-B-C	0.367	m3	0.4	1	5/10/2024	5/10/2024	1
62	Beam - 5-B-C	0.367	m3	0.4	1	5/10/2024	5/10/2024	1
63	Beam - 1-C-D	0.386	m3	0.4	1	5/11/2024	5/11/2024	1
64	Beam - 2-C-D	0.386	m3	0.4	1	5/11/2024	5/11/2024	1
65	Beam - 3-C-D	0.386	m3	0.4	1	5/12/2024	5/12/2024	1
66	Beam - 4-C-D	0.386	m3	0.4	1	5/12/2024	5/12/2024	1
67	Beam - 5-C-D	0.386	m3	0.4	1	5/13/2024	5/13/2024	1
68	Beam - 1-D-E	0.354	m3	0.4	1	5/13/2024	5/13/2024	1
69	Beam - 2-D-E	0.354	m3	0.4	1	5/14/2024	5/14/2024	1
70	Beam - 3-D-E	0.354	m3	0.4	1	5/14/2024	5/14/2024	1
71	Beam - 4-D-E	0.354	m3	0.4	1	5/15/2024	5/15/2024	1
72	Beam - 5-D-E	0.354	m3	0.4	1	5/15/2024	5/15/2024	1
73	Floor	607.42	m2	25.3	2	5/16/2024	5/27/2024	12
74	Beams & Floor Concrete	108.32	m3	6.8	8	5/28/2024	5/29/2024	2

Figure 4.39. The list of the tasks for the baseline schedule plan

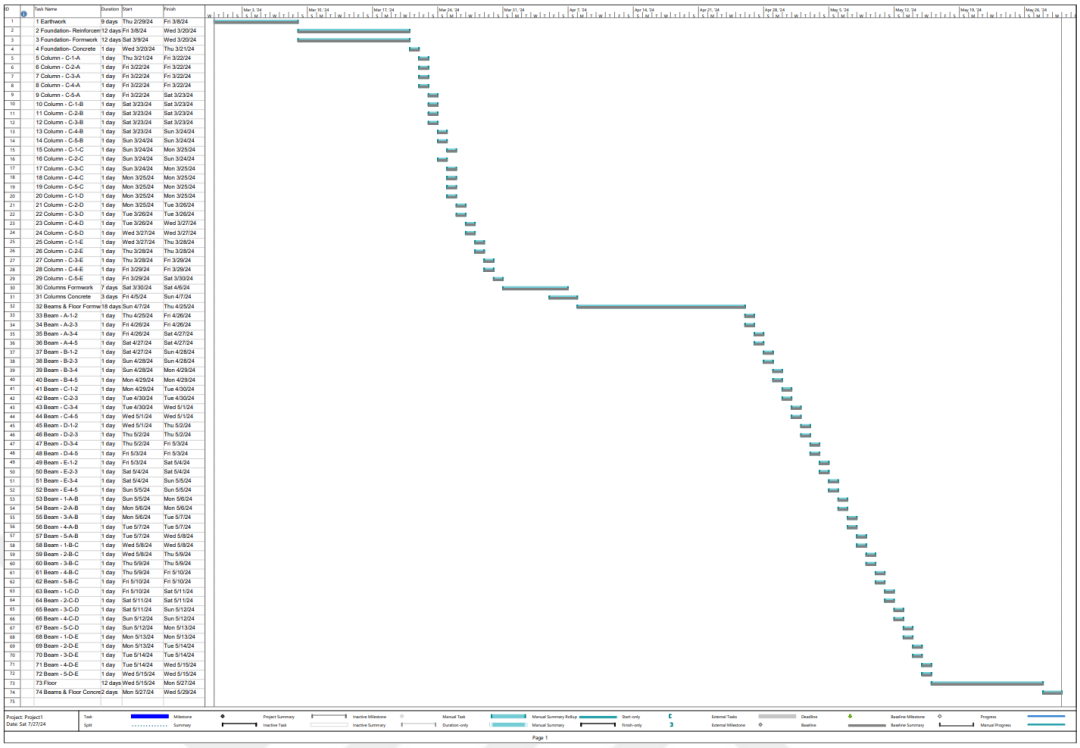


Figure 4.40. The baseline schedule plan Gantt chart

4.2.5 Results and Discussion

Following the simulation runs for three crashing modes, the baseline schedule plan is revised according to the space congestion and corresponding effect on the productivity of the agents. The revised schedule in comparison with the estimated expedited schedule plan reveals the impact of space congestion among the project crews on the project duration and cost and the importance of the quantitative analysis regarding the crashing analysis amid the construction stage of the project to remain within the time and cost borders.

The first mode signifies the increase in the concrete, rebar, and formwork resources from 1056 manhours to 1312 manhours aiming to shorten the project duration up to three weeks, while the second mode targets four weeks in duration decrease to meet the project end date at 01/05/2024 with the total manhour of 1362. Finally, the third

mode aims to achieve six weeks save regarding the project's overall duration by the resource usage of 1382 manhours.

On the other hand, the time-space conflicts detected by the developed simulation led to revisions for the simulation-based start and finish dates of the project activities revealing the gap between the estimated and quantitative crashing analysis. Figure 4.41 demonstrates the estimated and simulation-based schedule plans regarding crashing mode 1 where the end date of the project is postponed from 07/05/2024 to 17/05/2024 accordingly. Furthermore, Figure 4.42 illustrates the estimated and simulation-based schedule plans in crashing mode 2 indicating the end of the project on 10/05/2024 against the estimated end date of 01/05/2024 due to the space congestion. Similarly, Figure 4.43 illustrates the third mode targeting the end date of 20/04/2024 in comparison with the baseline schedule where the simulation-based schedule plan indicates the end date of 09/05/2024.

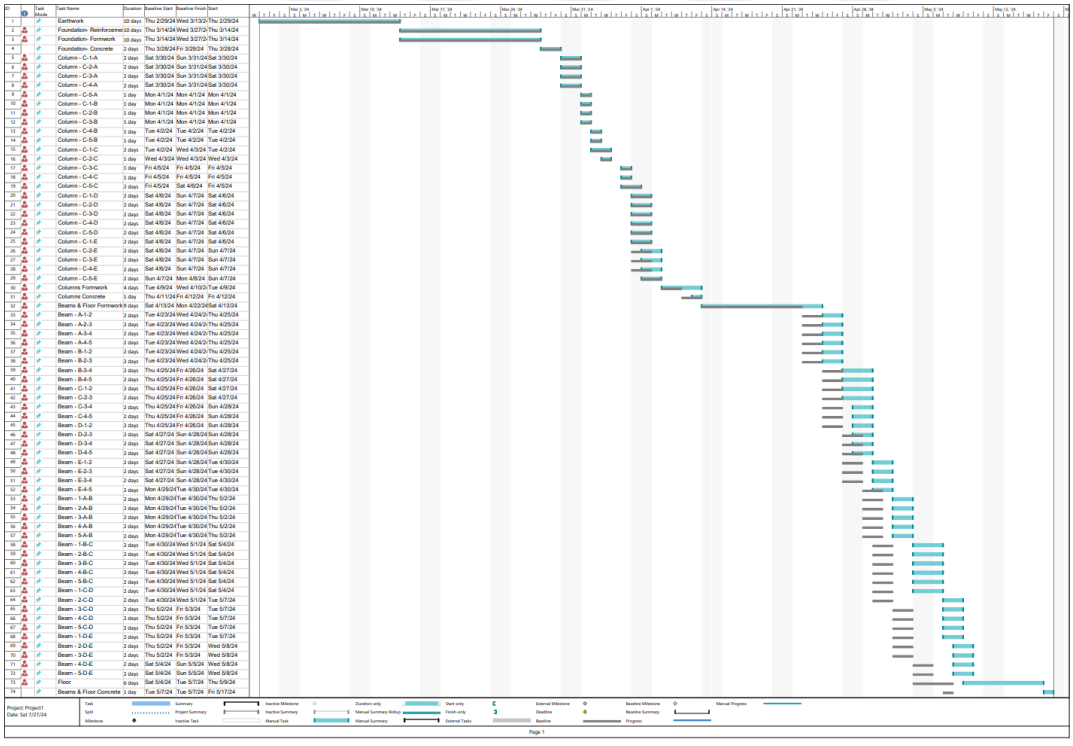


Figure 4.41. Estimated schedule plan (grey) vs simulation-based (light blue) schedule plan for crashing mode 1

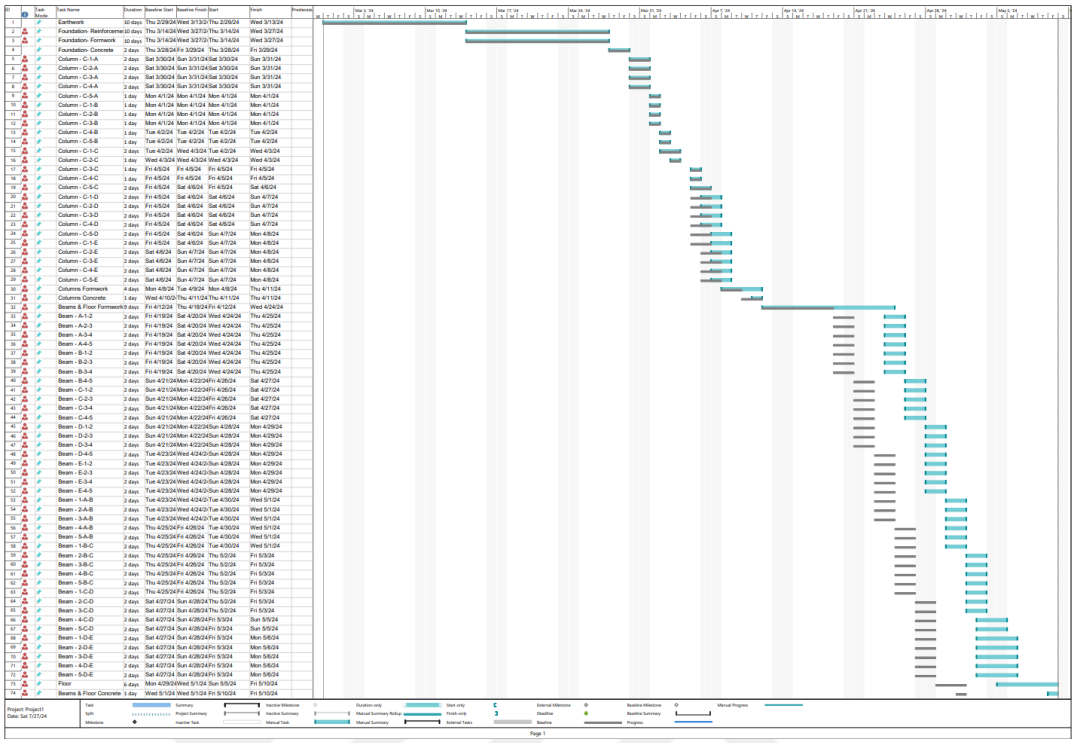


Figure 4.42. Estimated schedule plan (grey) vs simulation-based (light blue) schedule plan for crashing mode 2

To perform DTCTO analysis over the two mentioned modes, each mode's direct and indirect costs should be calculated. Table 4.4 indicates the time and cost associated with the crashing modes. The contractual clauses enforce a delay penalty of $900 for each day of delays considering the deadline of the project to be 20/04/2024. Figure 4.44 shows the DTCTO analysis considering the total cost including direct and indirect costs alongside the associated project duration.

Considering both time and cost notions, Mode-1 is dictated by Mode-2. Thus, Mode-1 leads to a non-optimal solution and should not be opted for unless there is a strict resource constraint. Mode-3 does not imply any improvement and does not lead to an optimal solution as the duration decrease is only one day taking into account the enforced additional resource cost. On the other hand, the crashing analysis with the traditional approach could indicate that Mode-3 is the best solution as it meets the project deadline and avoids the delay penalty accordingly. Consequently, it is

highlighted that the following simulation-based crashing analysis can have a determinative impact on managerial decisions in construction projects. Such decisions have a dominant effect on the time-cost balance of the project, leading to the final profit margins.

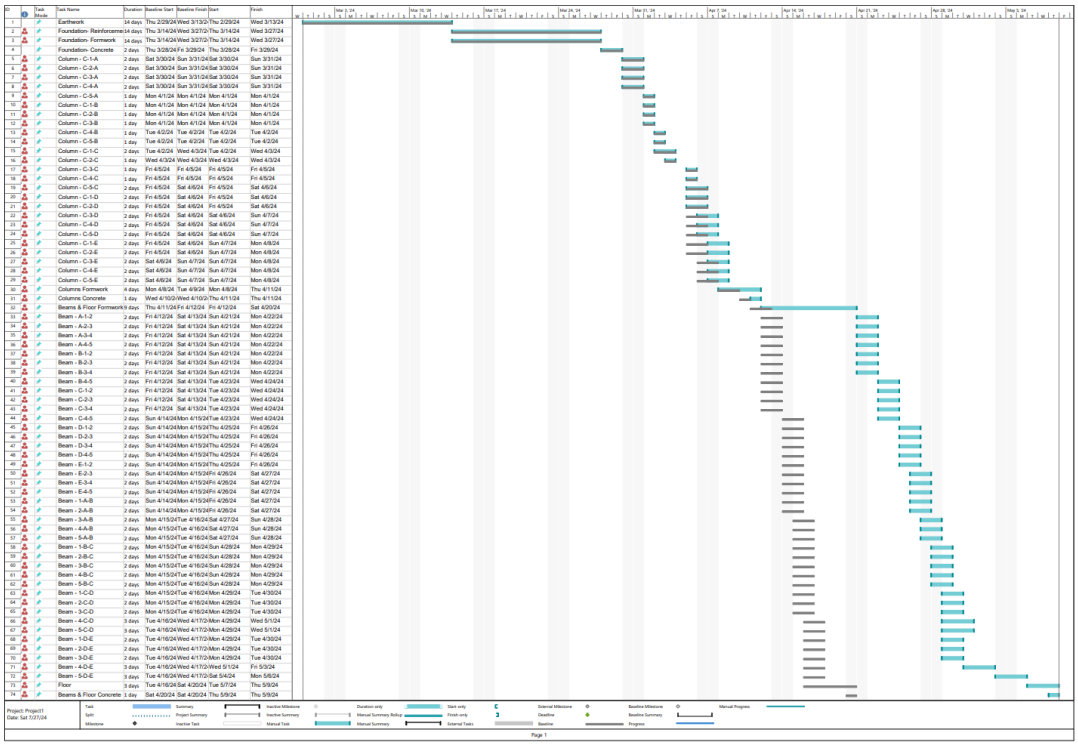


Figure 4.43. Estimated schedule plan (grey) vs simulation-based (light blue) schedule plan for crashing mode 3

Table 4.4 Crashing analysis considering the three modes

Crashing mode	Total manhour	Direct cost (48$ per hour)	Total project duration (days)
baseline	**1056**	**50,688**	**91**
one	1312	62,976	79
two	1362	65,376	72
three	1382	66,336	71

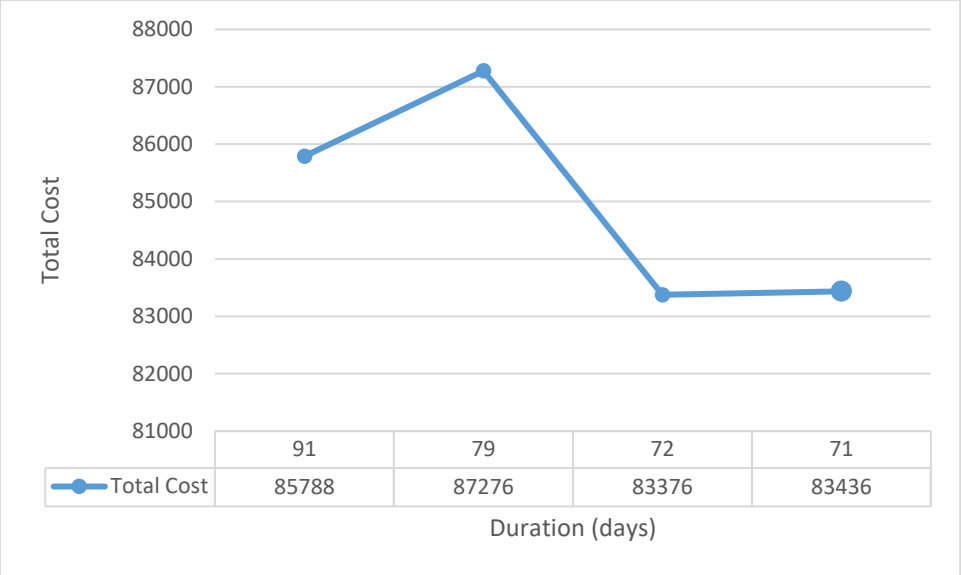


Figure 4.44. Crashing analysis to select the best crashing mode

Concerning the agent-level analyses, a selected task duration against the increase in the crew size for the rebar, formwork, and concrete cast works in baseline and simulation-based schedule plans is illustrated in Figure 4.45. In this graph, the effect of space congestion deteriorates the project duration decrease amount due to the productivity reduction. By increasing the crew size, the gap between the baseline and simulation-based schedule is widened more as represented in Figure 4.46.

4.2.6 Approach Benefits and Limitations

The implementation of the proposed method in Anylogic comes with a few limitations alongside the achieved benefits. The first limitation signifies the lowered fidelity in the simulation as the environment drops to 2D from the 3D sophisticated space in the gaming engine. This condition limits the visual validation of the simulation and hinders the simulation of complex spatial interactions that would be more accurately represented in a 3D environment.

Furthermore, the 2D environment in simulation affects the realism and immersion of the simulation limiting the insights derived from the simulation from the user's perspective.

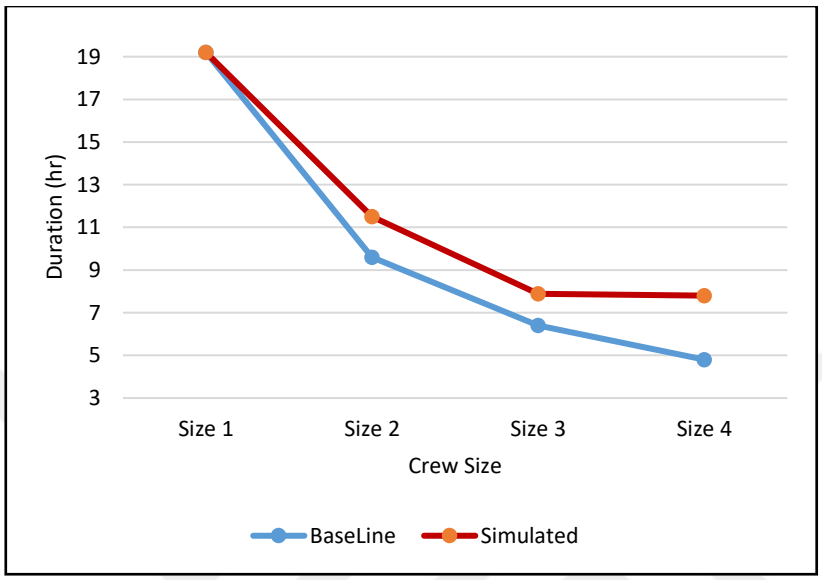


Figure 4.45. Agent-level task duration against increasing the crew size

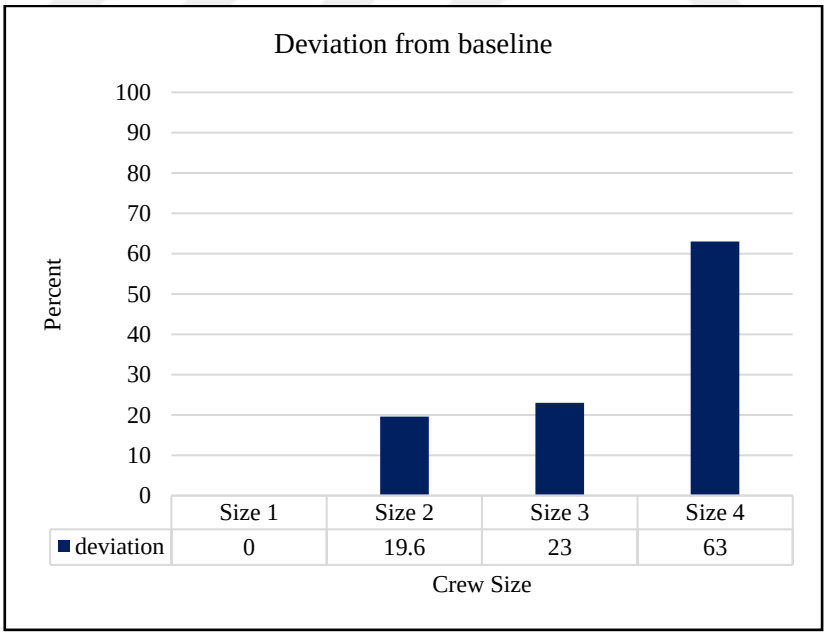


Figure 4.46. The gap between the baseline and simulation-based schedule plans by increasing the crew size

On the other hand, using Anylogic as a simulating tool provides robust simulation capabilities with external library integration that lowers the run duration considerably. The support of three different simulation types is another benefit allowing users to model complex scenarios by leveraging the strengths of each type.

CHAPTER 5

VERIFICATION & VALIDATION (V&V)

Verification and validation, also known as V&V, are two methods used for assuring the accuracy and reliability of a simulation model. Although they are occasionally used interchangeably, there are distinct differences between them. Verification aims to ensure that the model is built correctly, while validation seeks to confirm that the right model has been created. To achieve V&V in simulation models, two approaches can be followed: black box and white box. The black box approach focuses on the inputs and outputs of the simulation without relying on internal processes. In this approach, a set of parameters is selected for both the simulation and the real-world system, where the values of the parameters are compared between the simulation and the real-world system.

On the other hand, the white box approach concentrates on the internal processes and their reliability. This approach involves several methods but for the construction simulations, three methods of assessment of the chronological order of tasks, their duration, and the task cycle in the assigned resources are highlighted (Shi, 2002). In the current study, both black and white box approaches are applied to achieve verification and validation of the simulation. The chronological order of the tasks is assessed by referring to the recorder construction project video frames for the task execution and comparing them with the identical models created in the simulation environment. This approach aims to verify the simulation model. Concerning the validation, the simulation model with a set of predefined parameters is run and compared with the real-world estimated duration of the project with the same parameters.

5.1 Verification

The verification stands for the correctness of a product such as a simulation model considering its technical aspects that are specified in the first place prior to the development attempt. This step is an assertion that the simulation components are modeled correctly and that any misleading aspect is avoided accurately. In a simple statement, the verification stands for doing the model right (Shi, 2002). In the attempt for the verification of the proposed method in the current study, the chronological order of the tasks is evaluated in alignment with the white-box model verification approach.

The present study's chronological order of task reference relies on publicly accessible recorded construction site videos. These videos are primarily utilized as time-lapse overviews or as sources for tracking construction progress. The model presented in this study is assessed using real-world scenarios derived from the frames of these videos. The evaluation involves analyzing the frames to ensure the model's accuracy and relevance to actual construction activities. Detailed information about the video sources, including their respective links, can be found in Table 5.1. This comprehensive approach allows for a robust evaluation of the model, grounded in real-world data and practical applications.

Table 5.1 Video sources for V&V

Code	*Project Description*	*Link*
I	A steel building timelapse	click
II	A prefabricated steel building frame progress report	click
III	Erection of steel portal frames progress report part 1	click
IV	Erection of steel portal frames progress report part 2	click
V	A Steel building timelapse	click

The verification process involves the monitoring of the agent's behavior in executing a set of tasks, based on predefined patterns and checking if similar scenarios are

observed in video sources. This includes visually comparing video frames and setting up simulation scenes to illustrate the agent's actions in tasks such as material picking (lifting state), transportation to the destination (carrying state), and material installation (erection state) as depicted in Figure 5.1, Figure 5.2, Figure 5.3, Figure 5.4, and Figure 5.5. Additionally, projects associated with codes I and V are simulated under comparable conditions to enable a more comprehensive comparison with real-world cases. Consequently, the agent's behaviors and task sequencing are fine-tuned for all five real-world scenarios, ensuring that the new method can effectively simulate these scenarios.

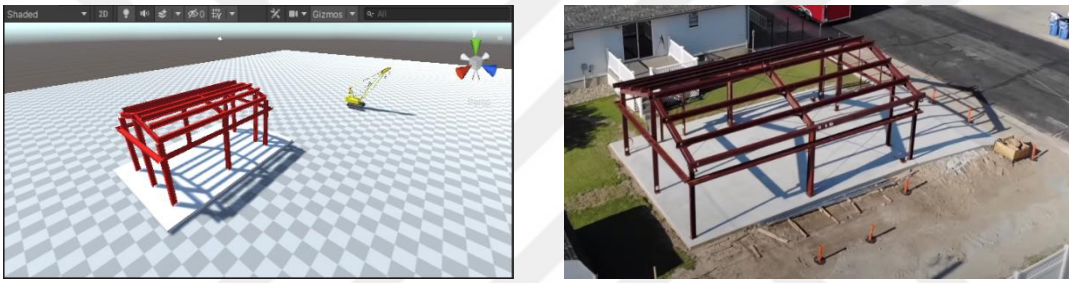


Figure 5.1. The model adapted from the video resource (Code I)

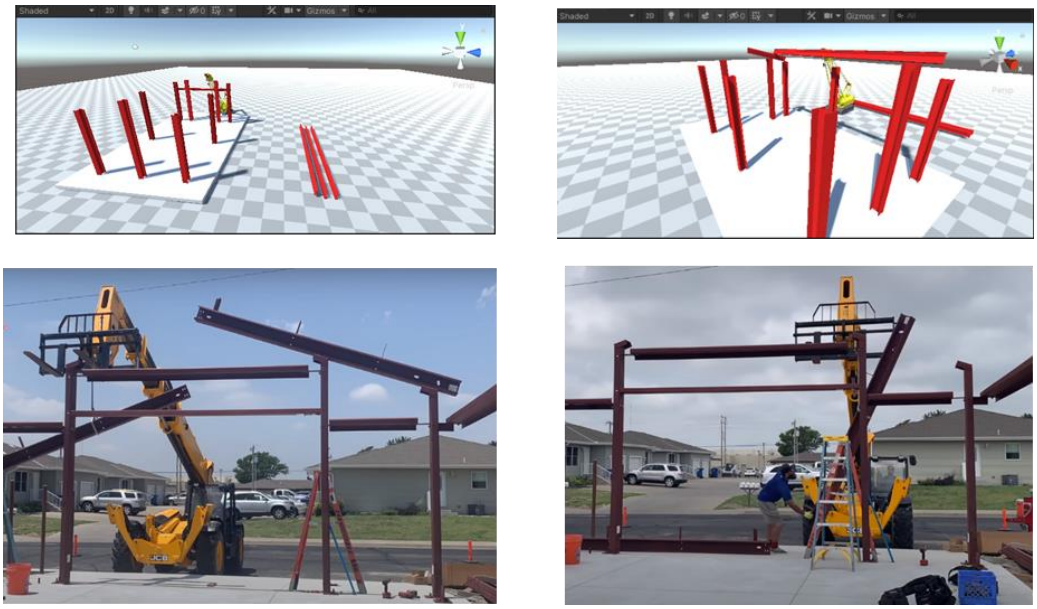


Figure 5.2. The behavior and sequence verification of the agent in comparison with the video frames in comparison with the modeled simulation - Code I

To evaluate the accuracy of the simulation in replicating real-world scenarios, we analyze the time it takes for an agent to complete a specific task based on their assigned productivity level. This time is then compared with the estimated task duration obtained from a reliable estimating database. Although the agent's productivity is initially defined as a user input within the simulation, we reference the RSMeans database to obtain a justified productivity value, which is then assigned to the agent. This process allows us to compare the duration of the steel structure erecting task within the simulation environment with the estimated duration from the database, helping us refine the model as needed to better reflect real-world conditions.



Figure 5.3. The sequence verification of the agent in comparison with the video frames - Code II



Figure 5.4. The sequence verification of the agent in comparison with the video frames - Code III and IV

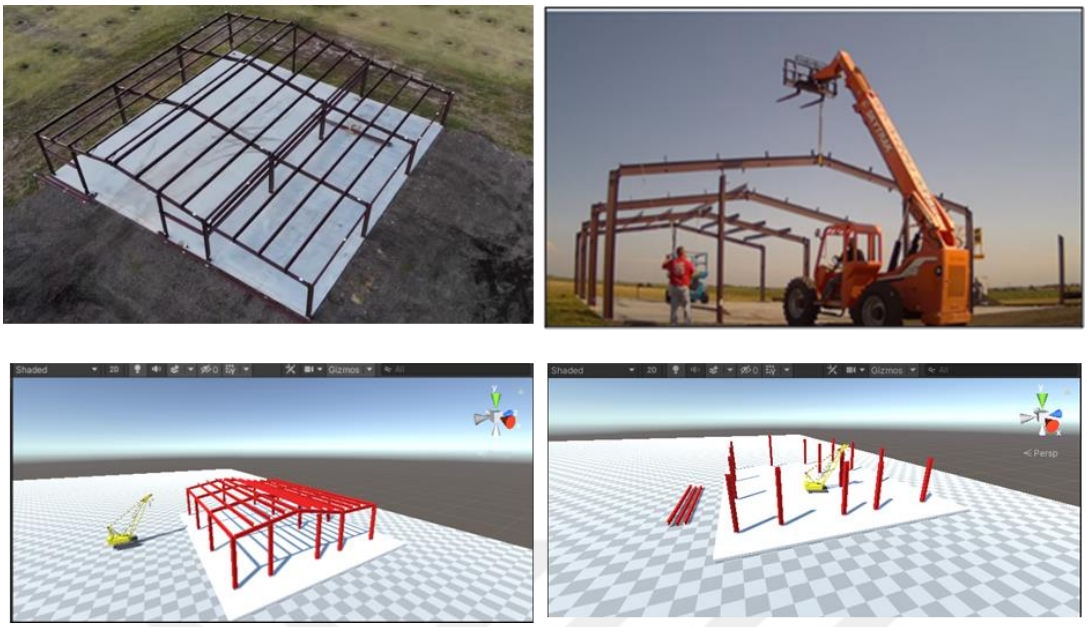


Figure 5.5. The behavior and sequence verification of the agent in comparison with the video frames in comparison with the modeled simulation - Code V

5.2 Validation

The validation is distinct from verification and is used to evaluate the correctness and reliability of the results in comparison to the real-world outputs. In other words, the validation is an attempt to answer if the right model is developed by appraising the pair of input-output in both model and real-world system. For this purpose, the black box approach is followed by comparing the simulation run with a predefined set of parameters with a well-known online estimation database namely the RSMeans database.

According to RSMeans data code 051223.77, steel structures with one to two stories have an estimated daily output of 10.30 tons. The labor required for this amount of work is projected to be 7.767 hours. Table 5.2 provides calculations for the crew's productivity and expected duration, which is approximately 44.51 working hours. The productivity values are used as parameters in a simulation run for a warehouse case project. The simulation results show that the duration for completing the steel

columns and beams is 46.83 working hours, closely matching the estimated value and demonstrating a good correlation.

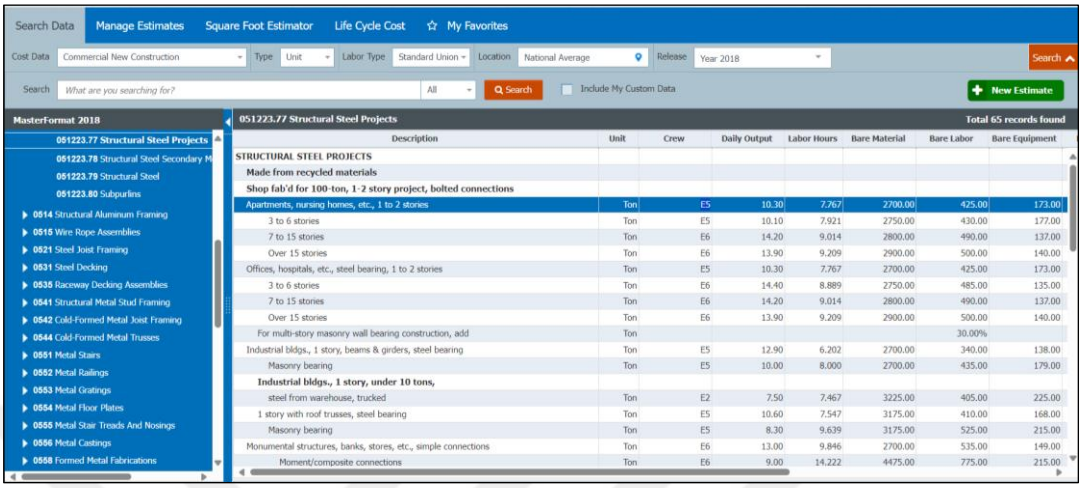


Figure 5.6. The estimated data retrieved from the RSMeans database

Table 5.2 Productivity and duration calculations based on the RSMeans database

Item	*Value*
Productivity obtained from RSMeans	10.30 tons per day
Total material weight in the case project	59.03 tons
Expected duration	$(59.03 \div 10.30) \times 7.767\ (labour\ hours)$ $= 44.51\ hours$
Estimated Productivity	$10.30 \div 7.767 = 1.326\ tons\ per\ hour$

CHAPTER 6

CONCLUSION

In the present study, a new method for generating and evaluating the crashing alternatives is proposed that is based on agent-based simulation (ABS) and BIM. It aims to increase the construction simulation fidelity and fulfill a part of the current gaps by quantifying time-space conflicts according to the crashing modes to support the time-cost trade-off and corresponding risk analyses. The method takes a 4D BIM model, an agent pool, and crashing alternatives as the input set and generates a simulation, in which the agents represent project manpower and machinery resources (crews) that can act independently on the basis of predefined behavioral patterns and move based on an artificial intelligence (AI) pathfinding algorithm. By benefiting the independent nature of the agents, a conflict avoidance mechanism that models the crews' behavior in real cases is also developed and integrated with them. In this mechanism, the agents dynamically assess the states of the other agents and make appropriate decisions aiming to avoid or minimize the conflicts.

The contribution of the study lies in filling some of the current gaps including oversimplification of crews' behavior in simulations and subjective approaches for crashing analysis by quantifying time-space conflicts and modeling distinct behaviors of project crews in order to generate crashing modes and obtain optimal crew size for each activity. The method aims to help project planners improve the accuracy of schedule compression estimations and foresee the emergent outcomes related to different time and cost scenarios by increasing the fidelity of construction simulations.

Furthermore, the modeling of execution patterns with respect to the tendency of the crews to reduce time-space conflicts during in-zone works by a multicriteria function. Additionally, to consider the uncertainties regarding the project parameters

and obtain corresponding outcomes, a probabilistic approach was integrated into the analyses. Using the method, in addition to contributing to the generation of quantified crashing modes, various analyses can be performed regarding the severity of time-space conflicts and their quantitative impact on the project performance, time-cost risk assessment, productivity values revision, in-path congestions, and construction method evaluation.

For future study, a comprehensive agent pool will be created, including various crews with different attributes, which can be usable in different projects with different construction methods. The method will be enhanced for implementing it on generic projects covering the entire work schedule and agent pool and will be developed to allow simultaneous optimization regarding the trio of time, cost, and space dimensions.

By using the proposed method, various achievements can be made, as listed below:

- Quantification of crashing modes
- Assessment of the severity of time-space conflicts and their impact on project performance
- Time-cost risk evaluations
- Revisions of productivity values
- Examination of in-path congestion
- Evaluation of construction methods
- Revisions of the schedule plans

In order to evaluate the capabilities of the suggested method, it is implemented in two environments: a game engine and a commercially available simulation tool. The game engine provides a sophisticated 3D environment with an effective integration with a BIM-based model. The Unity gaming engine is used in this case, and the model is created in Autodesk Revit and converted to a file format compatible with the engine. At the same time, a serverless database is tailored to the simulation model that is supplied with the model element data and the baseline schedule. The element

data is extracted from the model using a VLP called Dynamo and is integrated with the baseline schedule using Python coding language. The ABS simulation principles are implemented in the engine using C# programming language and .NET framework libraries. The built-in virtual sensory system in the engine enables the conflict detection mechanism between the simulation agents. Moreover, the simulation agent movement is developed using an AI pathfinding algorithm called A*. To implement the conflict avoidance behavior, additional codes are added to the algorithm accordingly.

In the second implementation approach, a commercially available simulation tool namely Anylogic is used. The tool is a hybrid simulation software supporting ABS, DES, and SD simulation types in an integrated environment. Considering the limitations of the tool, the model is integrated into a 2D environment using plan views of the model. Although this feature is a step back in the fidelity of the simulation, it boosts the simulation run speed and enables the involvement of more agents in a single run with a considerably small time span. The agents are generated with distinct behaviors represented by the statecharts. Moreover, the time-space conflict detection and conflict mechanism are implemented by functions developed in Java programming language. A built-in database is used to hold the model element data and record the simulation outputs. To involve the severe conflicts causing the constructability issues, a conflict matrix is added to the model. This matrix includes ones and zeros to distinguish the agents able to work in a congested space. The dynamic interaction of the agents with each other and the model elements results in simulation-based actual start and finish dates that are recorded in the database for further analysis. It should be noted that the model elements as physical obstacles are added to the model by the progress upon the project task execution.

The proposed BIM-integrated ABS method has a few limitations. Firstly, due to the highly detailed constituent parts of the simulation such as the 3D object-based environment, independent agents, and dynamic space analyses, the proposed method in the first implementation approach relies on the software and hardware capabilities, so the run speed cannot be increased upon a specific boundary. By exceeding this

threshold, the simulation could be prone to erroneous outputs due to the collapsing of the trigger and sensory systems, so the runs can take a considerably longer time than regular operational or 4D simulations. This drawback is avoided by implementing the method on a commercial simulation tool. However, this approach sacrifices the fidelity of the simulation dramatically as the environment should be converted to 2D. Therefore, a trade-off should be performed considering benefits and drawbacks in alignment with the requirements. Secondly, the definition of the zones and storage points and assigning them to the set of activities is up to the user amid the modeling stage which is a manual approach. Thirdly, in real cases, bearing in mind the uncertain and dynamic nature of space conflicts, it can be a big challenge to generate optimum solutions to come up with a congestion-free plan which was disregarded in the present study.

REFERENCES

AbouRizk, S., Halpin, D., Mohamed, Y., & Hermann, U. (2011). Research in Modeling and Simulation for Improving Construction Engineering Operations. Journal of Construction Engineering and Management, 137(10), 843–852. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000288

AbouRizk, S. M., & Halpin, D. W. (1990). Probabilistic Simulation Studies for Repetitive Construction Processes. Journal of Construction Engineering and Management, 116(4), 575–594. https://doi.org/10.1061/(ASCE)0733-9364(1990)116:4(575)

AbouRizk, S., & Mohamed, Y. (2002). Optimal construction project planning. Proceedings of the Winter Simulation Conference, 2, 1704–1708 vol.2. https://doi.org/10.1109/WSC.2002.1166454

Ahuja, H. N., & Nandakumar, V. (1985). Simulation Model to Forecast Project Completion Time. Journal of Construction Engineering and Management, 111(4), 325–342. https://doi.org/10.1061/(ASCE)0733-9364(1985)111:4(325)

Akinci, B., Fischen, M., Levitt, R., & Carlson, R. (2002). Formalization and Automation of Time-Space Conflict Analysis. Journal of Computing in Civil Engineering, 16(2), 124–134. https://doi.org/10.1061/(ASCE)0887-3801(2002)16:2(124)

Akinci, B., Fischer, M., & Kunz, J. (2002). Automated Generation of Work Spaces Required by Construction Activities. Journal of Construction Engineering and Management, 128(4), 306–315. https://doi.org/10.1061/(ASCE)0733-9364(2002)128:4(306)

Akinci, B., Fischer, M., & Zabelle, T. (1998). Proactive approach for reducing non-value adding activities due to time-space conflicts.

Akpan, I. J., Shanker, M., & Razavi, R. (2020). Improving the success of simulation projects using 3D visualization and virtual reality. Journal of the Operational Research Society, 71(12), 1900–1926. https://doi.org/10.1080/01605682.2019.1641649

Al-Hussein, M., Athar Niaz, M., Yu, H., & Kim, H. (2006). Integrating 3D visualization and simulation for tower crane operations on construction sites. Automation in Construction, 15(5), 554–562. https://doi.org/10.1016/j.autcon.2005.07.007

Alvanchi, A., Lee, S., & AbouRizk, S. (2011). Modeling Framework and Architecture of Hybrid System Dynamics and Discrete Event Simulation for

Construction. Computer-Aided Civil and Infrastructure Engineering, 26(2), 77–91. https://doi.org/10.1111/j.1467-8667.2010.00650.x

Arturo Osorio-Sandoval, C., Tizani, W., Pereira, E., Koch, C., & Fadoul, A. (2020). Discrete-Event Simulation and Building Information Modelling Based Animation of Construction Activities. https://doi.org/10.1007/978-3-030-51295-8_21

Attajer, A., & Mecheri, B. (2024). Multi-Agent Simulation Approach for Modular Integrated Construction Supply Chain. *Applied Sciences*, *14*(12), Article 12. https://doi.org/10.3390/app14125286

Arturo Osorio-Sandoval, C., Tizani, W., Pereira, E., Koch, C., & Fadoul, A. (2020). *Discrete-Event Simulation and Building Information Modelling Based Animation of Construction Activities*. https://doi.org/10.1007/978-3-030-51295-8_21

Aydoğan, E. B., & Atay, Y. (2021). Unity Based A* Algorithm Used in Shortest Path Finding Problem for Helicopters. 2021 International Conference on Control, Automation and Diagnosis (ICCAD), 1–5. https://doi.org/10.1109/ICCAD52417.2021.9638762

Ballesteros-Pérez, P., Elamrousy, K. M., & González-Cruz, M. C. (2019). Non-linear time-cost trade-off models of activity crashing: Application to construction scheduling and project compression with fast-tracking. Automation in Construction, 97, 229–240. https://doi.org/10.1016/j.autcon.2018.11.001

Bansal, V. K. (2011). Use of GIS and Topology in the Identification and Resolution of Space Conflicts. Journal of Computing in Civil Engineering, 25(2), 159–171. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000075

Ben-Alon, L., & Sacks, R. (2017). Simulating the behavior of trade crews in construction using agents and building information modeling. Automation in Construction, 74, 12–27. https://doi.org/10.1016/j.autcon.2016.11.002

Biswasa, S. K., Karmakera, C. L., & Biswasa, T. K. (2016). Time-cost trade-off analysis in a construction project problem: Case study. *International Journal of Computational Engineering Research (IJCER)*, *6*(10), 32-38.

Borshchev, A. (2014). Multi-method modelling: AnyLogic. In Discrete-Event Simulation and System Dynamics for Management Decision Making (pp. 248–279). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118762745.ch12

Borshchev, A., Karpov, Y., & Kharitonov, V. (2002). Distributed simulation of hybrid systems with AnyLogic and HLA. Future Generation Computer Systems, 18(6), 829–839. https://doi.org/10.1016/S0167-739X(02)00055-9

Chavada, R., Dawood, N., & Kassem, M. (2012). Construction workspace management: The development and application of a novel nD planning approach and tool. Journal of Information Technology in Construction (ITcon), 17(13), 213–236.

Choi, B., Lee, H.-S., Park, M., Cho, Y. K., & Kim, H. (2014). Framework for Work-Space Planning Using Four-Dimensional BIM in Construction Projects. Journal of Construction Engineering and Management, 140(9), 04014041. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000885

Chua, D. K. H., Yeoh, K. W., & Song, Y. (2010). Quantification of Spatial Temporal Congestion in Four-Dimensional Computer-Aided Design. Journal of Construction Engineering and Management, 136(6), 641–649. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000166

Cui, X., & Shi, H. (2010). A*-based Pathfinding in Modern Computer Games. 11.

Dabirian, S., Moussazadeh, M., Khanzadi, M., & Abbaspour, S. (2023). Predicting the effects of congestion on labour productivity in construction projects using agent-based modelling. International Journal of Construction Management, 23(4), 606–618. https://doi.org/10.1080/15623599.2021.1901330

Dashti, M. S., RezaZadeh, M., Khanzadi, M., & Taghaddos, H. (2021). Integrated BIM-based simulation for automated time-space conflict management in construction projects. Automation in Construction, 132, 103957. https://doi.org/10.1016/j.autcon.2021.103957

Dawood, N., & Mallasi, Z. (2006). Construction Workspace Planning: Assignment and Analysis Utilizing 4D Visualization Technologies. Computer-Aided Civil and Infrastructure Engineering, 21(7), 498–513. https://doi.org/10.1111/j.1467-8667.2006.00454.x

Ding, Z., Gong, W., Li, S., & Wu, Z. (2018). System Dynamics versus Agent-Based Modeling: A Review of Complexity Simulation in Construction Waste Management. Sustainability, 10(7), Article 7. https://doi.org/10.3390/su10072484

Dozzi, S. P., & AbouRizk, S. M. (1993). *Productivity in construction* (p. 54). Ottawa: Institute for Research in Construction, National Research Council.Eastman, C., & Others, A. (1974). An Outline of the Building Description System. Research Report No. 50. https://eric.ed.gov/?id=ED113833

Eastman, C. (1975). The use of computers instead of drawings in building design. *AIA journal*, *63*(3), 46-50.

Fathi, S., Fathi, S., & Balali, V. (2023). Time–Space Conflict Management in Construction Sites Using Discrete Event Simulation (DES) and Path

Planning in Unity. Applied Sciences, 13(14), Article 14. https://doi.org/10.3390/app13148128

Feng, C.-W., Liu, L., & Burns, S. A. (2000). Stochastic Construction Time-Cost Trade-Off Analysis. Journal of Computing in Civil Engineering, 14(2), 117–126. https://doi.org/10.1061/(ASCE)0887-3801(2000)14:2(117)

Getuli, V., Capone, P., Bruttini, A., & Isaac, S. (2020). BIM-based immersive Virtual Reality for construction workspace planning: A safety-oriented approach. Automation in Construction, 114, 103160. https://doi.org/10.1016/j.autcon.2020.103160

Guo, S.-J. (2002). Identification and Resolution of Work Space Conflicts in Building Construction. Journal of Construction Engineering and Management, 128(4), 287–295. https://doi.org/10.1061/(ASCE)0733-9364(2002)128:4(287)

Halpin, D. W. (1973). *An investigation of the use of simulation networks for modeling construction operations*. University of Illinois at Urbana-Champaign.

H, V. D. Parunak. (1998). Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. Multi-Agent Systems and Agent-Based Simulation, 10. https://cir.nii.ac.jp/crid/1571417126678968064

Helbing, D., & Balietti, S. (2012). How to Do Agent-Based Simulations in the Future: From Modeling Social Mechanisms to Emergent Phenomena and Interactive Systems Design.

Hemalattha, S., & Vidjeapriya, R. (2021). Developing an integrated framework for optimization of spatial requirements of construction equipment. Built Environment Project and Asset Management, 11(5), 903–917. https://doi.org/10.1108/BEPAM-07-2020-0126

Hong, Z., Shi, J. J., & Tam, C. M. (2002). Visual modeling and simulation for construction operations. Automation in Construction, 11(1), 47–57. https://doi.org/10.1016/S0926-5805(01)00049-8

Hosny, A., Nik-Bakht, M., & Moselhi, O. (2020). Workspace planning in construction: Non-deterministic factors. Automation in Construction, 116, 103222. https://doi.org/10.1016/j.autcon.2020.103222

Huang, A. L., Chapman, R. E., & Butry, D. T. (2009). Metrics and Tools for Measuring Construction Productivity: Technical and Empirical Considerations. NIST. https://www.nist.gov/publications/metrics-and-tools-measuring-construction-productivity-technical-and-empirical

Indhu, B., & Ajai, P. (2014). Study of delay management in a construction project-a case study. *International Journal of Emerging Technology and Advanced Engineering*, *4*(5), 108-113.

Ioannou, P. G. (1990). UM-CYCLONE Discrete Event Simulation System User's Guide, (1998). P.G. Ioannou, UM-CYCLONE Discrete Event Simulation System User's Guide, (1998).

Junjie, X., Jie, Z., Jiyang, X., & Weijie, K. (2023). Research on Time-Space-Frequency Multi-Domain Conflict Detection Method for UAV Cluster Mission Planning. *2023 12th International Conference of Information and Communication Technology (ICTech)*, 464–468. https://doi.org/10.1109/ICTech58362.2023.00093

Kamat, V. R., & Martinez, J. C. (2001). Visualizing Simulated Construction Operations in 3D. Journal of Computing in Civil Engineering, 15(4), 329–337. https://doi.org/10.1061/(ASCE)0887-3801(2001)15:4(329)

Kassem, M., Dawood, N., & Chavada, R. (2015). Construction workspace management within an Industry Foundation Class-Compliant 4D tool. Automation in Construction, 52, 42–58. https://doi.org/10.1016/j.autcon.2015.02.008

Khataei, S., Akcamete, A., & Sonmez, R. (2020). 4D BIM integrated productivity estimation of construction projects. In *6th International Project and Construction Management Conference (IPCMC2020)*.

Khodabandelu, A., & Park, J. (2021). Agent-based modeling and simulation in construction. Automation in Construction, 131, 103882. https://doi.org/10.1016/j.autcon.2021.103882

Kim, K., & Kim, K. J. (2010). Multi-agent-based simulation system for construction operations with congested flows. Automation in Construction, 19(7), 867–874. https://doi.org/10.1016/j.autcon.2010.05.005

Kuntz, K. A. (1994). A construction crew evaluation model to improve craft productivity [Ph.D., The Pennsylvania State University]. https://www.proquest.com/docview/304134869/abstract/439531215CFE48B8PQ/1

K.W., Y., David K H, C., & Associate Professor, Department of Civil Engineering, National University of Singapore. (2012). Mitigating Workspace Congestion: A Genetic Algorithm Approach. Proceedings of the 2012 (3rd) International Conference on Engineering, Project, and Production Management, 107–118. https://doi.org/10.32738/CEPPM.201209.0011

Liu, B., Yang, B., Zhang, B., Dong, M., Jiang, S., & Xiao, J. (2022). Spatio-temporal deduction of floor construction based on the agent modeling of construction actors. Automation in Construction, 142, 104487. https://doi.org/10.1016/j.autcon.2022.104487

Lu, M. (2003). Simplified Discrete-Event Simulation Approach for Construction Simulation. Journal of Construction Engineering and Management, 129(5), 537–546. https://doi.org/10.1061/(ASCE)0733-9364(2003)129:5(537)

Luo, J. (2024). System Dynamics-Based Simulation Analysis of the Policy of the Prefabricated Building Industry Chain. *Tehnički Vjesnik*, *31*(3), 982–992. https://doi.org/10.17559/TV-20230315000439

Lutz, J. D., Halpin, D. W., & Wilson, J. R. (1994). Simulation of Learning Development in Repetitive Construction. Journal of Construction Engineering and Management, 120(4), 753–773. https://doi.org/10.1061/(ASCE)0733-9364(1994)120:4(753)

Macal, C. M. (2010). To agent-based simulation from System Dynamics. Proceedings of the 2010 Winter Simulation Conference, 371–382. https://doi.org/10.1109/WSC.2010.5679148

Mallasi, Z. (2006). Dynamic quantification and analysis of the construction workspace congestion utilising 4D visualisation. Automation in Construction, 15(5), 640–655. https://doi.org/10.1016/j.autcon.2005.08.005

Maria, A. (1997). Introduction to modeling and simulation. Proceedings of the 29th Conference on Winter Simulation - WSC '97, 7–13. https://doi.org/10.1145/268437.268440

Marschall, M., Tahmasebi, F., & Burry, J. (2019). Including occupant behavior in building simulation: Comparison of a deterministic vs. a stochastic approach. Proceedings of the Symposium on Simulation for Architecture and Urban Design, 1–4.

Martinez, J. C., & Ioannou, P. G. (1997, October). State-based probabilistic scheduling using STROBOSCOPE's CPM add-on. In *Proc., Construction Congress V* (pp. 438-445). Reston, Va.: ASCE.

Mastli, M., & Zhang, J. (2017). Interactive Highway Construction Simulation Using Game Engine and Virtual Reality for Education and Training Purpose. 399–406. https://doi.org/10.1061/9780784480830.049

Mawlana, M., Vahdatikhaki, F., Doriani, A., & Hammad, A. (2015). Integrating 4D modeling and discrete event simulation for phasing evaluation of elevated urban highway reconstruction projects. Automation in Construction, 60, 25–38. https://doi.org/10.1016/j.autcon.2015.09.005

Messi, L., García de Soto, B., Carbonari, A., & Naticchia, B. (2022). Spatial conflict simulator using game engine technology and Bayesian networks for workspace management. Automation in Construction, 144, 104596. https://doi.org/10.1016/j.autcon.2022.104596

Mirzaei, A., Nasirzadeh, F., Parchami Jalal, M., & Zamani, Y. (2018). 4D-BIM Dynamic Time–Space Conflict Detection and Quantification System for

Building Construction Projects. Journal of Construction Engineering and Management, 144(7), 04018056. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001504

Moghadam, M., Al-Hussein, M., Al-Jibouri, S., & Telyas, A. (2012). Post simulation visualization model for effective scheduling of modular building construction1This paper is one of a selection of papers in this Special Issue on Construction Engineering and Management. Canadian Journal of Civil Engineering, 39(9), 1053–1061. https://doi.org/10.1139/l2012-077

Moon, H., Kim, H., Kim, C., & Kang, L. (2014). Development of a schedule-workspace interference management system simultaneously considering the overlap level of parallel schedules and workspaces. Automation in Construction, 39, 93–105. https://doi.org/10.1016/j.autcon.2013.06.001

Naciri, L., Gallab, M., Soulhi, A., Merzouk, S., & Nardo, M. D. (2024). Modeling and simulation: A comparative and systematic statistical review. Procedia Computer Science, 232, 242–253. https://doi.org/10.1016/j.procs.2024.01.024

Nazarko, J., & Chodakowska, E. (2015). Measuring Productivity of Construction Industry in Europe with Data Envelopment Analysis. Procedia Engineering, 122, 204–212. https://doi.org/10.1016/j.proeng.2015.10.026

Özgün, O., & Barlas, Y. (2009, July). Discrete vs. continuous simulation: When does it matter. In *Proceedings of the 27th international conference of the system dynamics society* (Vol. 6, pp. 1-22).

Patacas, J. M. D. L., Dawood, N., & Kassem, M. (2014). Evaluation of IFC and COBie as data sources for asset register creation and service life planning. In *14th International Conference on Construction Applications of Virtual Reality*.

Pérez, C. T., Fernandes, L. L. A., & Costa, D. B. (2016). A LITERATURE REVIEW ON 4D BIM FOR LOGISTICS OPERATIONS AND WORKSPACE MANAGEMENT. Proc. 24th Ann. Conf. of the Int'l. Group for Lean Construction, Boston, MA, USA, 53–62.

Petridis, P., Dunwell, I., de Freitas, S., & Panzoli, D. (2010). An Engine Selection Methodology for High Fidelity Serious Games. 2010 Second International Conference on Games and Virtual Worlds for Serious Applications, 27–34. https://doi.org/10.1109/VS-GAMES.2010.26

Pradhananga, N., & Teizer, J. (2014). Congestion Analysis for Construction Site Layout Planning using Real-Time Data and Cell-Based Simulation Model. 681–688. https://doi.org/10.1061/9780784413616.085

Project Management Institute (Ed.). (2003). A guide to the project management body of knowledge: (PMBOK® Guide (2000th ed.). Project Management Institute.

Riley, D. R., & Sanvido, V. E. (1995). Patterns of Construction-Space Use in Multistory Buildings. Journal of Construction Engineering and Management, 121(4), 464–473. https://doi.org/10.1061/(ASCE)0733-9364(1995)121:4(464)

Riley, D. R., & Sanvido, V. E. (1997). Space Planning Method for Multistory Building Construction. Journal of Construction Engineering and Management, 123(2), 171–180. https://doi.org/10.1061/(ASCE)0733-9364(1997)123:2(171)

Robinson, S. (2005). Discrete-event simulation: From the pioneers to the present, what next? Journal of the Operational Research Society, 56(6), 619–629. https://doi.org/10.1057/palgrave.jors.2601864

Said, H., & El-Rayes, K. (2013). Optimal utilization of interior building spaces for material procurement and storage in congested construction sites. Automation in Construction, 31, 292–306. https://doi.org/10.1016/j.autcon.2012.12.010

Sanders, S. R. (1998). An analysis of factors affecting labor productivity in masonry construction [Ph.D., The Pennsylvania State University]. https://www.proquest.com/docview/303717510/abstract/2014324428054AE4PQ/1

Sawhney, Bashford, Walsh, & Mulky. (2003). Agent-based modeling and simulation in construction. Proceedings of the 2003 Winter Simulation Conference, 2003., 2, 1541–1547 vol.2. https://doi.org/10.1109/WSC.2003.1261600

Semenov, V., Anichkin, A., Morozov, S., Tarlapan, O., & Zolotov, V. (2014). Effective Project Scheduling Under Workspace Congestion and Workflow Disturbance Factors. Australasian Journal of Construction Economics and Building - Conference Series, 2(1), Article 1. https://doi.org/10.5130/ajceb-cs.v2i1.3768

Shi, J. J. (2002). Three methods for verifying and validating the simulation of a construction operation. *Construction Management & Economics, 20*(6), 483-491.

Sterman, J. (2002). System Dynamics: Systems Thinking and Modeling for a Complex World [Working Paper]. Massachusetts Institute of Technology. Engineering Systems Division. https://dspace.mit.edu/handle/1721.1/102741

Tak, A. N., Taghaddos, H., Mousaei, A., Bolourani, A., & Hermann, U. (2021). BIM-based 4D mobile crane simulation and onsite operation management. Automation in Construction, 128, 103766. https://doi.org/10.1016/j.autcon.2021.103766

Tang, V., & Vijay, S. (2001). System dynamics. Origins, Development, and Future Prospects of a Method.

https://stuff.mit.edu/afs/sipb/project/wine/dosdevices/z:/afs/athena/course/other/esd.83/www/notebook/System%20Dynamics%20final.doc

Tantisevi, K., & Akinci, B. (2007). Automated generation of workspace requirements of mobile crane operations to support conflict detection. Automation in Construction, 16(3), 262–276. https://doi.org/10.1016/j.autcon.2006.05.007

Tantisevi, K., & Akinci, B. (2008). Simulation-Based Identification of Possible Locations for Mobile Cranes on Construction Sites. Journal of Computing in Civil Engineering, 22(1), 21–30. https://doi.org/10.1061/(ASCE)0887-3801(2008)22:1(21)

Tao, S., Wu, C., Sheng, Z., & Wang, X. (2018). Space-Time Repetitive Project Scheduling Considering Location and Congestion. Journal of Computing in Civil Engineering, 32(3), 04018017. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000745

Thabet, W. Y., & Beliveau, Y. J. (1994). Modeling Work Space to Schedule Repetitive Floors in Multistory Buildings. Journal of Construction Engineering and Management, 120(1), 96–116. https://doi.org/10.1061/(ASCE)0733-9364(1994)120:1(96)

Thabet, W. Y., & Beliveau, Y. J. (1997). SCaRC: Space-Constrained Resource-Constrained Scheduling System. Journal of Computing in Civil Engineering, 11(1), 48–59. https://doi.org/10.1061/(ASCE)0887-3801(1997)11:1(48)

Thomas, H. R., Riley, D. R., & Sinha, S. K. (2006). Fundamental Principles for Avoiding Congested Work Areas—A Case Study. Practice Periodical on Structural Design and Construction, 11(4), 197–205. https://doi.org/10.1061/(ASCE)1084-0680(2006)11:4(197)

Wang, Q., Curcuruto, M., Mei, Q., Liu, S., Zhou, Q., & Zhang, J. (2019). Agent-Based Modeling of Employee Protection-Oriented Safety Proactivity Behaviors at Small Scale Enterprises. BioMed Research International, 2019, e2471418. https://doi.org/10.1155/2019/2471418

Younes, A., & Marzouk, M. (2018). Tower cranes layout planning using agent-based simulation considering activity conflicts. Automation in Construction, 93, 348–360. https://doi.org/10.1016/j.autcon.2018.05.030

Zaman, A. A. U., Abdelaty, A., & Sobuz, M. H. R. (2024). Integration of BIM data and real-time game engine applications: Case studies in construction safety management. *Journal of Information Technology in Construction (ITcon)*, *29*(7), 117–140. https://doi.org/10.36680/j.itcon.2024.007

Zhang, J. P., & Hu, Z. Z. (2011). BIM- and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during

construction: 1. Principles and methodologies. Automation in Construction, 20(2), 155–166. https://doi.org/10.1016/j.autcon.2010.09.013

Zhang, P., Li, N., Jiang, Z., Fang, D., & Anumba, C. J. (2019). An agent-based modeling approach for understanding the effect of worker-management interactions on construction workers' safety-related behaviors. Automation in Construction, 97, 29–43. https://doi.org/10.1016/j.autcon.2018.10.015

Zhang, S., Teizer, J., Pradhananga, N., & Eastman, C. M. (2015). Workforce location tracking to model, visualize and analyze workspace requirements in building information models for construction safety planning. Automation in Construction, 60, 74–86. https://doi.org/10.1016/j.autcon.2015.09.009

APPENDICES

A. Case One Simulation Run Screenshots

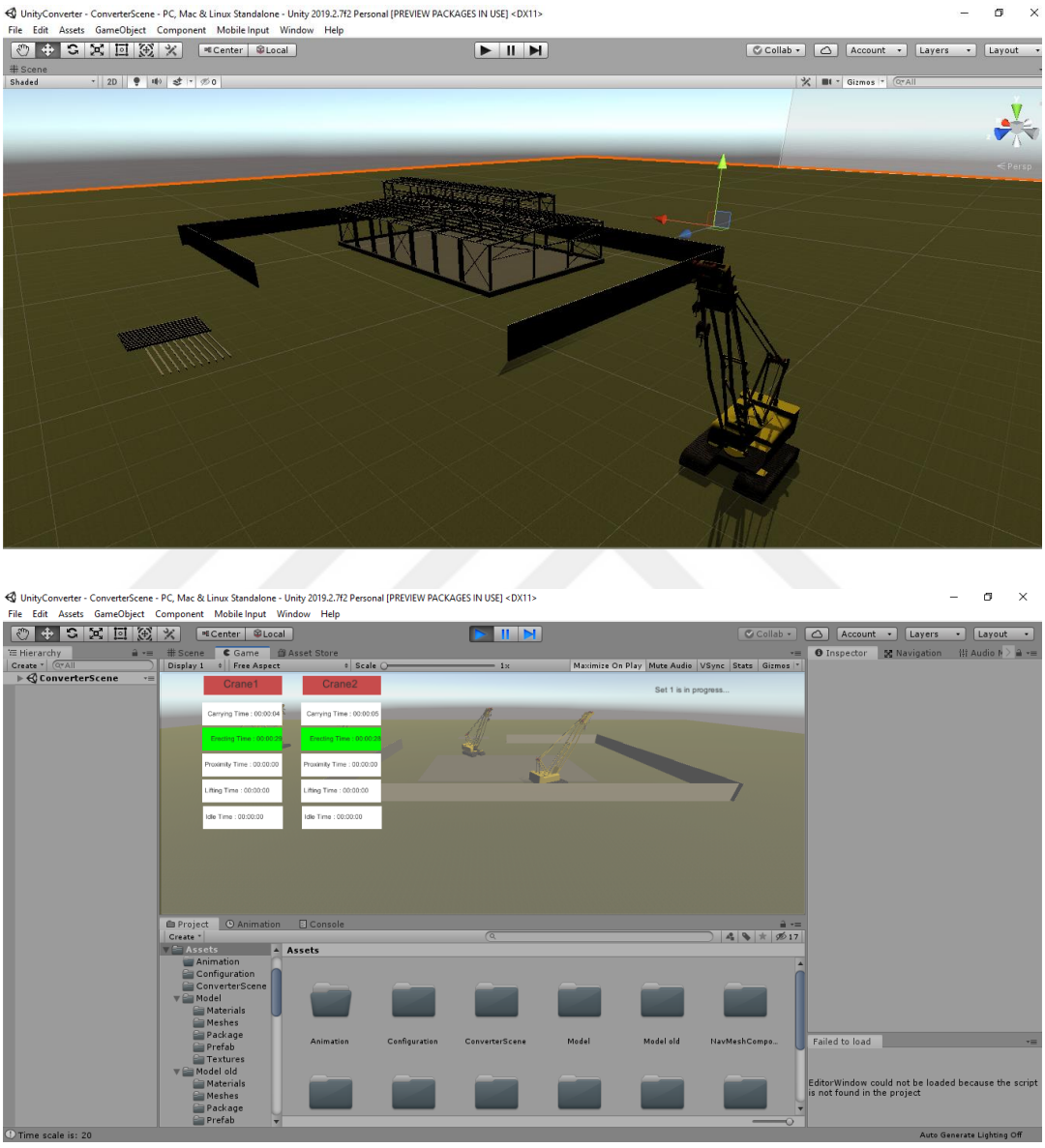


Figure 6.1. A simulation run scene in Unity environment

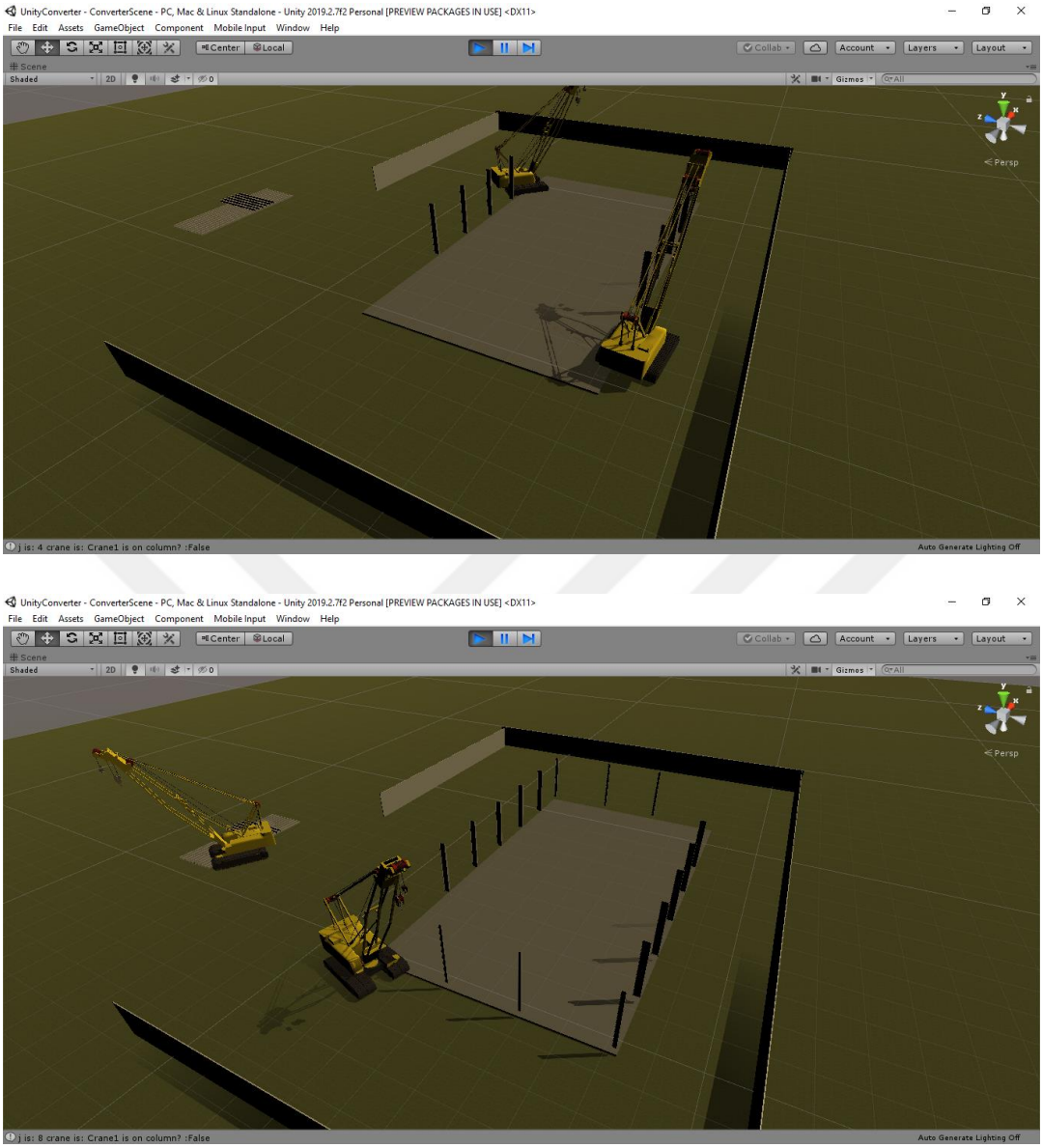


Figure 6.2. A scene of a simulation run in Unity indicating two agents mounting the steel elements

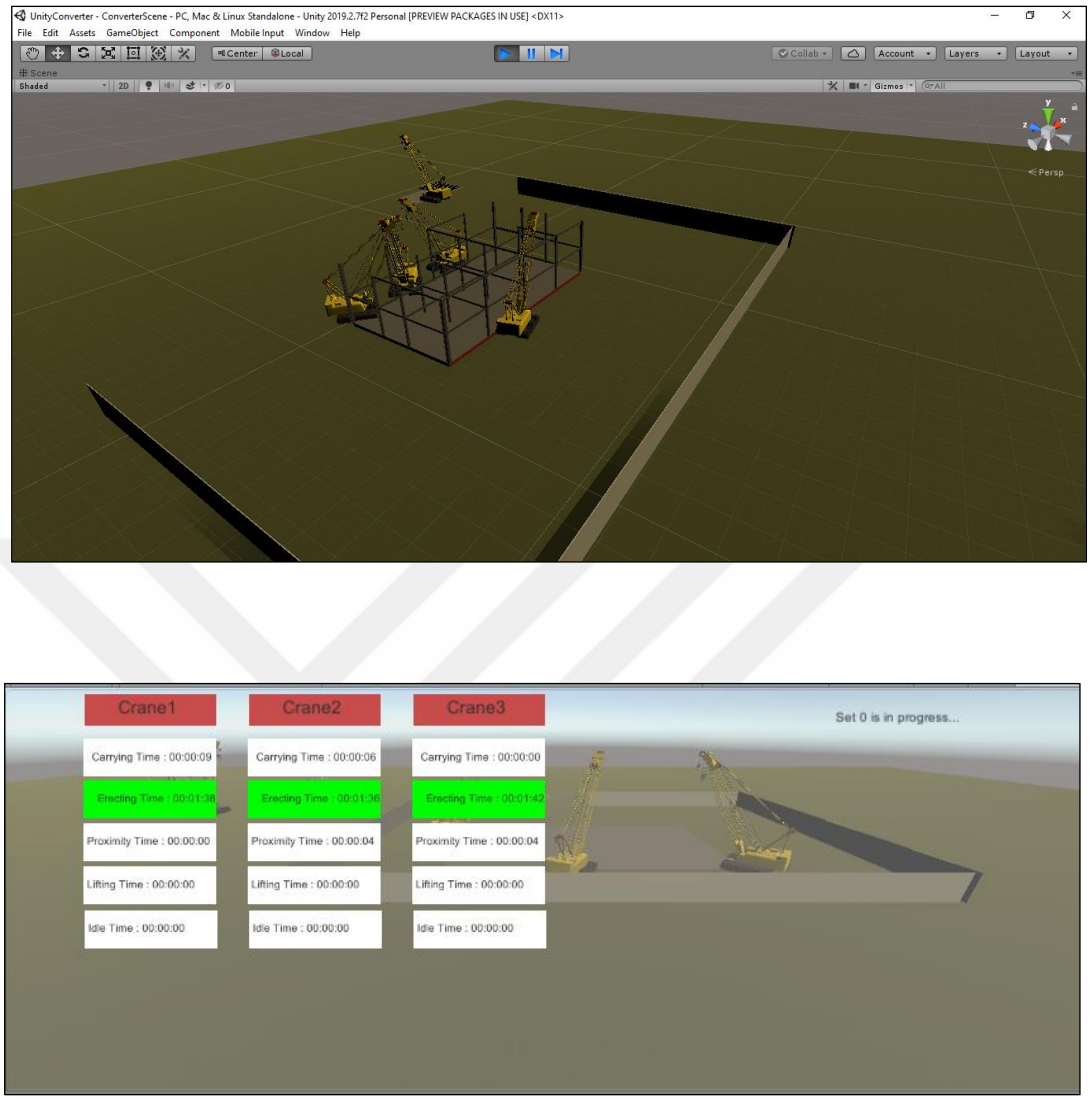


Figure 6.3. A simulation run scene in Unity with more than two agents

B. Case Two Simulation Run Screenshots

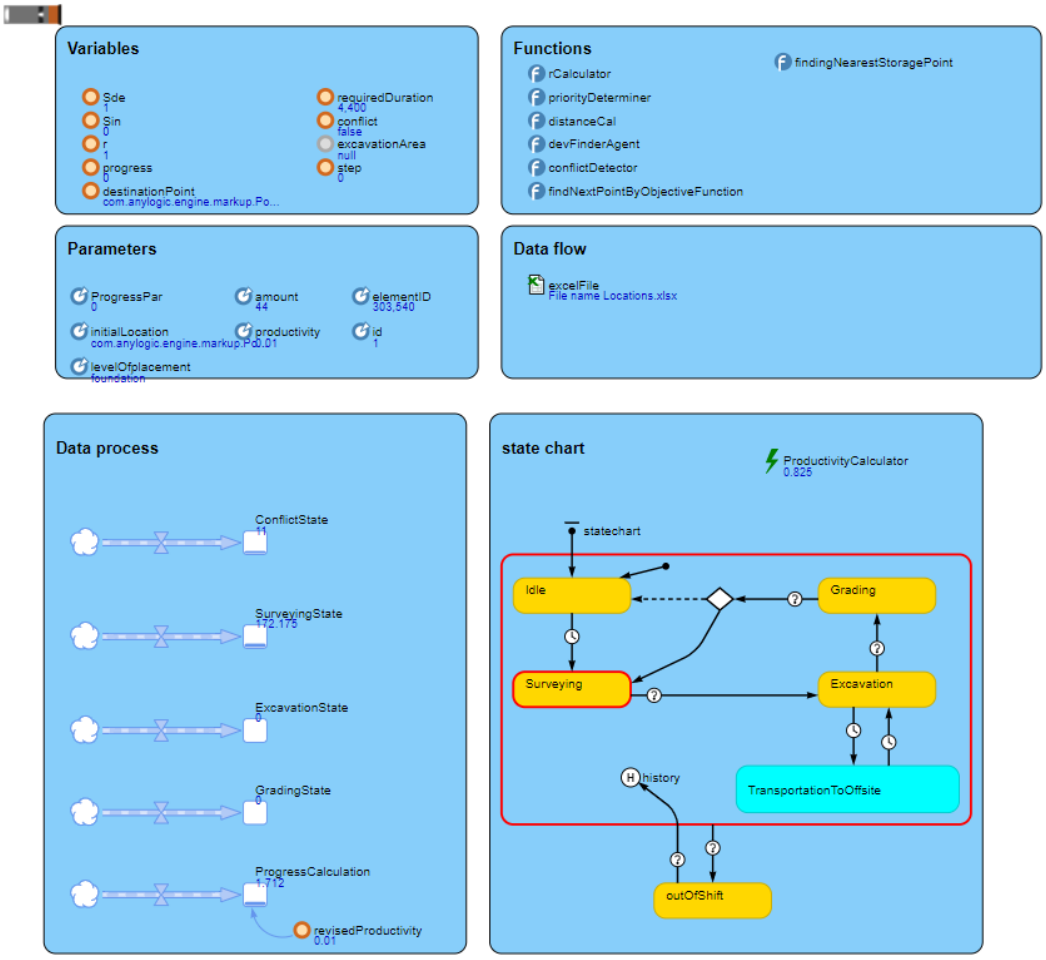


Figure 6.4. The excavator agent development environment in Anylogic

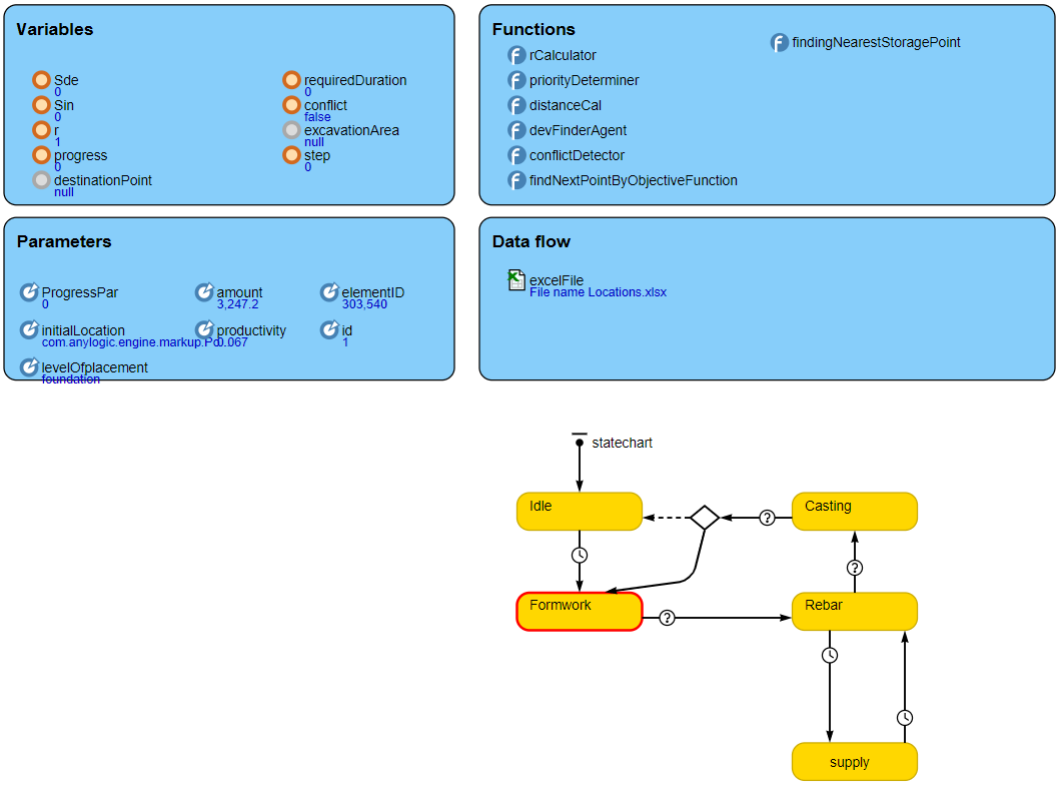


Figure 6.5. The concrete agent development environment in Anylogic



Figure 6.6. The simulation landing page environment for adjusting the conflict matrix in Anylogic

id	activity	amount	unit	location	points	agent	size	productivity_per_working_hour	relationship
1	Earthwork	220	m3	foundation	1	excavator	5	3	-
2	Foundation- Reinforcement	16,236	kg	foundation	10	concreteCrew	5	20	F.S - 2
3	Foundation- Formwork	98.6	m	foundation	8	concreteCrew	3	5	F.S - 3
4	Foundation- Concrete	220	m3	foundation	2	concreteCrew	3	30	F.S - 4
5	Column - C-1-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4
6	Column - C-2-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4
7	Column - C-3-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4
8	Column - C-4-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 4
9	Column - C-5-A	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9
10	Column - C-1-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9
11	Column - C-2-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9
12	Column - C-3-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 9
13	Column - C-4-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12
14	Column - C-5-B	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12
15	Column - C-1-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12
16	Column - C-2-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 12
17	Column - C-3-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16
18	Column - C-4-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16
19	Column - C-5-C	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16
20	Column - C-1-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 16
21	Column - C-2-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20
22	Column - C-3-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20
23	Column - C-4-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20
24	Column - C-5-D	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 20
25	Column - C-1-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24
26	Column - C-2-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24
27	Column - C-3-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24
28	Column - C-4-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 24
29	Column - C-5-E	0.8	m3	level 1	1	concreteCrew	2	0.2	F.S - 28
30	Columns Formwork	160	m2	level 1	15	concreteCrew	4	10	F.S - 29
31	Columns Concrete	12	m3	level 1	15	concreteCrew	4	1.5	F.S - 30
32	Beams & Floor Formwork	758.27	m2	level 2	40	concreteCrew	4	10	F.S - 31

Figure 6.7. A view of the used database schema in Anylogic

CURRICULUM VITAE

Surname, Name: Khataei, Saman

EDUCATION

Degree	Institution	Year of Graduation
MS	TABRIZ University - Earthquake Engineering	2014
BS	TABRIZ AZAD University - Civil Engineering	2011
High School	Ehsan High School, Tabriz	2010

PROFESSIONAL EXPERIENCE

Role	Location	Experience (Years)
Project Estimator	Tabriz	1
Schedule Engineer	Tabriz	5
Schedule Manager	Tabriz	2
Project Manager	Tabriz/ Tehran	2
BIM Manager	Remote	2
Simulation Engineer	Remote	2
BIM Expert	Ankara	Continues

ACADEMIC EXPERIENCE

Role	Location	Experience (Years)
Project Assistant	METU	2
Teaching Assistant	METU	4

FOREIGN LANGUAGES

Advanced English, Fluent Turkish, Fluent Farsi, Native Azeri

CODING LANGUAGES

C# (Proficient), Java (Proficient), Python (Mid-level), Dynamo VLP (Proficient), Javascript (Proficient)

PUBLICATIONS

1. Khataei, S., Akcamete, A., & Sonmez, R. A BIM-Integrated Agent-Based Simulation Method for Time-Space Conflict Detection Among Mobile Resources in Construction Projects (Submitted to Automation in Construction).

2. Khataei, S., Akcamete, A., & Sonmez, R. (2020). 4D BIM integrated productivity estimation of construction projects. In 6th International Project and Construction Management Conference (IPCMC2020- Istanbul/Turkey).

3. Khataei, S., Akçamete, A., & Sonmez, R. (2018). An Assessment of BIM-Based Construction Quantity Take-Offs (IPCMC2018- Cyprus).

TEZ İZİN FORMU / THESIS PERMISSION FORM

ENSTİTÜ / INSTITUTE

Fen Bilimleri Enstitüsü / Graduate School of Natural and Applied Sciences ■

Sosyal Bilimler Enstitüsü / Graduate School of Social Sciences ☐

Uygulamalı Matematik Enstitüsü / Graduate School of Applied Mathematics ☐

Enformatik Enstitüsü / Graduate School of Informatics ☐

Deniz Bilimleri Enstitüsü / Graduate School of Marine Sciences ☐

YAZARIN / AUTHOR

Soyadı / Surname : KHATAEI
Adı / Name : SAMAN
Bölümü / Department : İnşaat Mühendislik / Civil Engineering

TEZİN ADI / TITLE OF THE THESIS (**İngilizce** / English) : A BIM-INTEGRATED AGENT-BASED SIMULATION METHOD FOR TIME-SPACE CONFLICT DETECTION AMONG MOBILE RESOURCES IN CONSTRUCTION PROJECTS

TEZİN TÜRÜ / DEGREE: **Yüksek Lisans** / Master ☐ **Doktora** / PhD ■

1. **Tezin tamamı dünya çapında erişime açılacaktır. /** Release the entire work immediately for access worldwide. ■
2. **Tez iki yıl süreyle erişime kapalı olacaktır.** / Secure the entire work for patent and/or proprietary purposes for a period of **two year. *** ☐
3. **Tez altı ay süreyle erişime kapalı olacaktır.** / Secure the entire work for period of **six months**. ***** ☐

***** *Enstitü Yönetim Kurulu Kararının basılı kopyası tezle birlikte kütüphaneye teslim edilecektir.*
A copy of the Decision of the Institute Administrative Committee will be delivered to the library together with the printed thesis.

Yazarın imzası / Signature ..... **Tarih** / Date 29.09.2024