CCC 2024

Proceedings of the Creative Construction Conference (2024)
Edited by: Miroslaw J. Skibniewski & Miklós Hajdu
https://doi.org/10.3311/CCC2024-098

Automating Bridge Construction Scheduling data with BIM and Machine Learning

Israel Simon Mariaca Clavel[1]
Laura Florez-Perez[2]
Carlos Galera Zarco[3]

[1] University College London, London WC1E 6BT, UK, israel.mariaca.20@ucl.ac.uk
[2] University College London, London WC1E 6BT, UK, l.florez@ucl.ac.uk
[3] University College London, London WC1E 6BT, UK, c.galerazarco@ucl.ac.uk

Abstract

Conventional construction scheduling techniques often fall short due to inefficiencies, the prevalence of human error, and the lack of reliable scheduling data. Despite efforts to obtain better scheduling data and automate the process using Building Information Modelling (BIM) as a primary source of information, challenges persist. BIM models contain inconsistencies that can compromise their scheduling effectiveness. Moreover, existing automation efforts face obstacles, including complexity and scalability issues. These efforts frequently overlook the intricacies of accurately modelling construction tasks, neglecting resource considerations, subprocesses, and structured information management, which further complicates the creation of reliable and efficient construction schedule data. This research proposes a framework designed to systematically obtain scheduling data in an automated manner, aligning with a Work Breakdown Structure (WBS) to ensure organisational clarity. By incorporating subprocesses and focusing on resource constraint calculations and balancing. Furthermore, the framework structures scheduling data into a CSV format, facilitating easier analysis and integration with project management tools. The research follows the Design Science Research methodology. This multi-stage framework integrates BIM data extraction, automated element labelling, creation of a custom WBS, construction sequence, and resource balancing. The framework was developed using Python coding and libraries, facilitating seamless transitions between stages without manual intervention. The framework's performance is evaluated using 4D BIM software, assessing the generated data and logic in a girder bridge projects. This innovative framework enhances the accuracy and efficiency of construction data scheduling through automation. It reduces manual intervention, organises data effectively, and improves project timeline reliability. The use of 4D BIM further illustrates the practical application of this data in bridge projects, showcasing a scalable and robust contribution to construction automation.

Keywords: Scheduling; Game-Based Simulation; Building Information Modelling (BIM); Real-Time Strategy (RTS) Mechanics; Game Engine; Dynamic; Simulation.

1. Introduction

Construction scheduling encompasses several high-level steps fundamental across various scheduling techniques. These core steps include task identification, which enumerates all necessary tasks for the project; task sequencing, establishing the order and dependencies among tasks; and task duration estimation, determining the time required to complete each task [1].

Historically, construction scheduling has been predominantly a manual effort, relying heavily on specific expertise and experience. While this traditional approach is grounded in practical knowledge, it is inherently susceptible to human errors and inefficiencies [2]. Recognizing these challenges, the construction industry has witnessed numerous attempts to automate the scheduling process. These efforts aim to transcend the limitations of manual scheduling by leveraging technological advancements. Automated scheduling, in this context, refers to the implementation of computer-assisted tools designed to enhance and optimize the construction scheduling process [3]. Many attempts to automate the construction scheduling have been made through the use of Building Information Modelling (BIM). The study conducted by Sheikhkhoshkar [4] utilised a 4D Building Information Management technique to create an automated solution for concrete joint location. Although the attempts to use BIM models for automated scheduling showed some solutions, there are some drawbacks, mainly concerning the type and quality of the construction data for the automation processes extracted from BIM models [5].

This research focuses on creating a framework that generates scheduling data from BIM models in an automated manner. Following the Design Science Research methodology, this study aims first to understand the current limitations and context of the literature, then define the objectives of the solution, design and develop the framework, and finally evaluate it through the use of a 4D BIM simulation, a simple girder bridge was used to showcase the framework process and the same project was validated in the 4D BIM simulation. The paper is organized as follows: introduction, literature review and context, design and development of the framework, validation, and conclusion.

2. Literature review and context

2.1 Automate scheduling

As defined in the literature, "Automated scheduling in this context refers to the use of computer-assisted tools to streamline and optimize the scheduling process in the construction industry" [6]. Advancements in scheduling automation are characterized by diverse methodologies, each bringing unique impacts. The Graph-based Schedule Mining GSM approach, as detailed in [7], emphasizes capturing tacit knowledge to streamline schedule generation. However, it overlooks off-site activities and project risks. Conversely, [8] employs 3D modelling for detailed, flexible scheduling but necessitates intricate setup and high-quality models. The application of 3D modelling represents a notable advancement, offering detailed and flexible planning solutions. However, there is a recognized requirement for the development and refinement of scheduling data generation methods applicable across diverse construction scheduling approaches.

2. BIM in scheduling automation; Data extraction and labelling

BIM has emerged as a cornerstone for the automation of the scheduling process, serving as the primary source of information [9]. The process entailed systematic data extraction from BIM models, [10] reinforced the significance of BIM in generating realistic construction schedules by integrating varied planning data. This study introduced an automated method to define interdependencies between construction activities, traditionally a manual and error-prone process, using templates based on linked building information data. Furthermore, [11] proposed an innovative framework using BIM to align the Engineering Breakdown Structure (EBS) of BIM with the traditional Work Breakdown Structure (WBS) of construction schedules. Despite its forward-thinking approach, the study acknowledged limitations like the inability to automatically estimate activity durations and its dependency on the quality and completeness of BIM model. Across these studies, a shared limitation is evident in the data extraction and element labelling processes within BIM models. Specifically, the studies highlight challenges in accurately extracting and categorizing detailed information from BIM models, an issue compounded by the varying complexity of projects and inconsistencies in data formats across different BIM software.

2.3 Task sequencing, task time calculation and resource balancing

In contemporary research on construction sequencing, various methodologies have been investigated. [5] used the WBS to organize and group the building elements, which in turn influenced the construction sequence. However, the effectiveness of the proposed integrated information model and data integration process relied on the availability and quality of data from BIM model. [12] utilizes a BIM-based Multi-Objective Genetic Algorithm for construction task sequencing, focusing on schedule optimization. A common limitation observed across these studies is the challenge of integrating complex BIM data with scheduling algorithms. Research efforts encounter difficulties related to data complexity and scalability, underscoring the challenge of effectively utilizing detailed BIM inputs to produce efficient and practical scheduling outcomes.

Considering the inherent limitations encountered in automating construction scheduling via BIM models, a gap has been identified. This gap pertains to the transition between data extracted from BIM models and its application within automated scheduling techniques, necessitating preliminary data preprocessing. Considering this, this research proposes a novel framework designed to facilitate the automation of data extraction and preprocessing from BIM models. This framework aims to automate the generation of construction scheduling data that can be used for performing scheduling calculations.

3. Framework

The research framework integrates several interrelated processes designed to leverage BIM for construction automation data, as depicted in **Figure 1**. The framework consists of four primary stages: BIM model data extraction, automated element labelling, data processing and resource balancing. Each stage is interconnected and plays a crucial role in achieving an automated and efficient construction scheduling data, Central to the integration and seamless transition between these stages is the use of Python as the coding language for the development of each process. By utilizing Python's extensive libraries and its ability to handle data-intensive tasks, each stage has been encoded to allow for smooth linking and integration, ensuring that data flows effectively from one phase to the next without the need for manual intervention. The framework emphasizes the fluid integration between each phase, with dashed lines indicating the framework's progression and solid lines representing the data flow. The input data, processes, and output are clearly delineated, ensuring clarity and ease of understanding for stakeholders involved in the construction automation process.

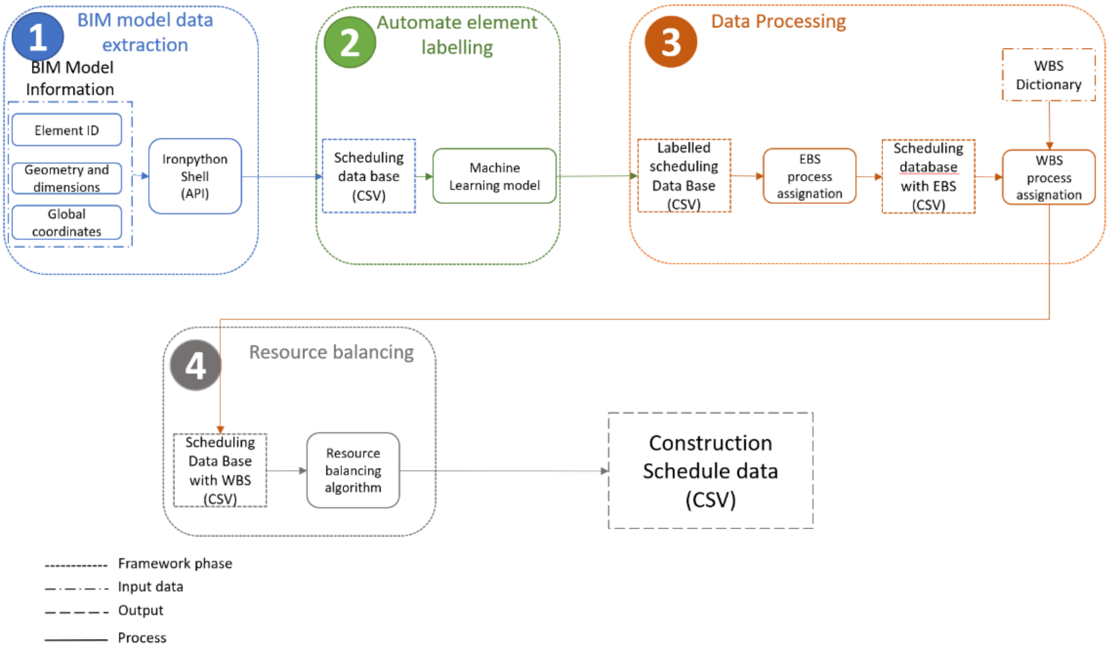
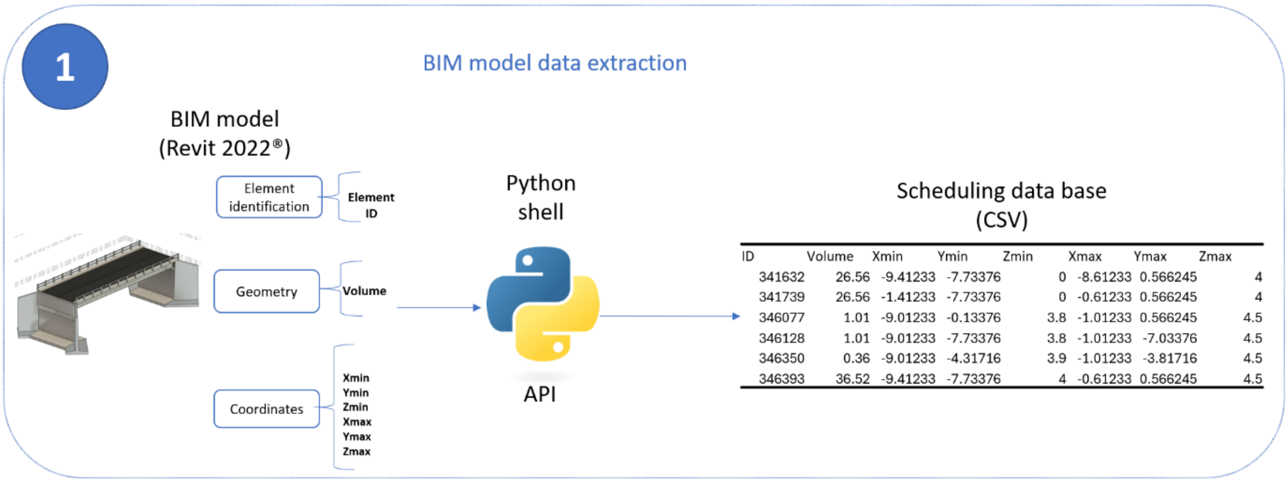


Figure 1 Proposed framework.

3.1 BIM model data extraction

BIM models generally serve as a pivotal source of comprehensive metadata, playing an instrumental role in project scheduling. The BIM model, specifically from Revit 2022®, contains rich information encapsulated within each model element which is essential for accurate and efficient planning. **Figure 2** illustrates the process flow for the data extraction phase.



ID	Volume	Xmin	Ymin	Zmin	Xmax	Ymax	Zmax
341632	26.56	-9.41233	-7.73376	0	-8.61233	0.566245	4
341739	26.56	-1.41233	-7.73376	0	-0.61233	0.566245	4
346077	1.01	-9.01233	-0.13376	3.8	-1.01233	0.566245	4.5
346128	1.01	-9.01233	-7.73376	3.8	-1.01233	-7.03376	4.5
346350	0.36	-9.01233	-4.31716	3.9	-1.01233	-3.81716	4.5
346393	36.52	-9.41233	-7.73376	4	-0.61233	0.566245	4.5

Figure 2 BIM model data extraction.

To facilitate the extraction of key metadata from the BIM model, a custom Application Programming Interface (API) was developed that interfaces with a Python shell. This API is designed to automatically export the necessary data into a CSV file format. For element identification, we extract only the element ID from the BIM model. The volume of each BIM element is extracted as it directly correlates to the quantity of work to be performed. Additionally, the coordinates of each element are extracted and included in the dataset. These spatial parameters are vital for the precise labelling and positioning of elements in later stages, particularly during the automated element labelling, where spatial relationships and sequencing are key factors.

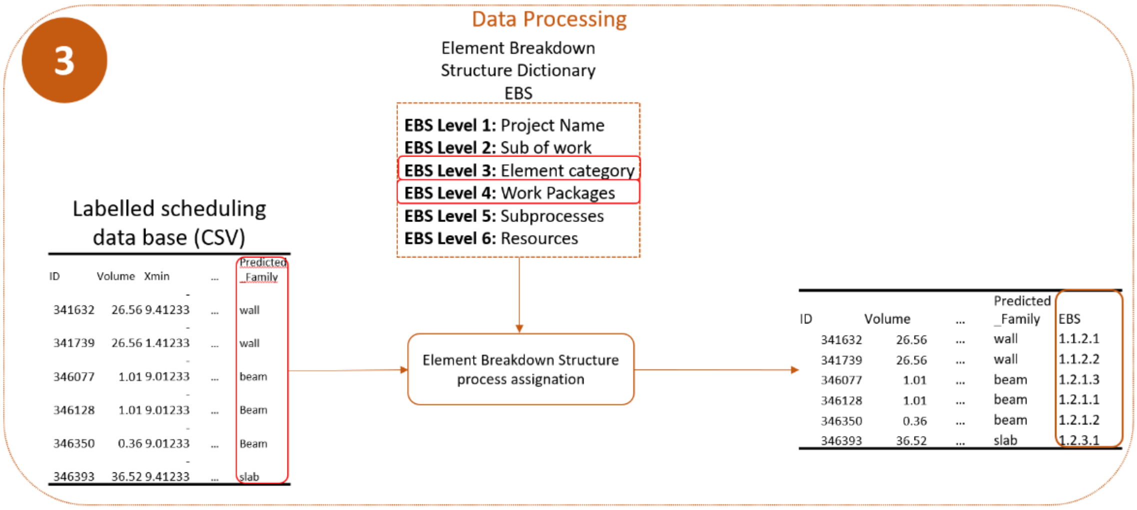
3.2 Automate labelling of the construction elements using Machine Learning

Automated labelling of construction elements using machine learning techniques presents a promising avenue for improving the efficiency and accuracy of construction scheduling [13]. This phase outlines the process of applying a RandomForest Classifier, due to its ability to handle high-dimensional data, robustness to overfitting, and scalability, to the information extracted from the BIM model, aiming to automate the task of labelling based on their geometric properties and global coordinates. The initial phase of this phase involves the preprocessing of training datasets which contains various attributes of construction elements, including their coordinates and volumetric information. The preprocessing steps included the calculation of each element's width, depth, and height by subtracting their respective minimum and maximum X, Y, and Z coordinates. Following preprocessing, we selected relevant features for the machine learning model, focusing on the geometric attributes of each construction element: minimum and maximum coordinates (Xmin, Xmax, Ymin, Ymax, Zmin, Zmax), width, depth, height, and volume.

These features were used as inputs (X) to predict the 'Type' of each element, serving as the output variable (y). The dataset was then split into training and test subsets, employing a random seed for reproducibility. A RandomForest Classifier was trained using the training subset, chosen for its robustness and ability to handle the complexity of the dataset without extensive parameter tuning. Additionally, the RandomForest Classifier was serialized and saved using the pickle library, facilitating the reuse of the trained model for future labelling tasks without the need for retraining. After loading the trained model, the new dataset is pre-processed in the same manner as the training dataset to calculate necessary features. These predictions, representing the model's classification of each element into predefined categories, were added to the dataset as a new column, "Predicted_Family" Finally, the enriched dataset, now containing both original attributes and predicted labels, was saved to a CSV file. This step ensures the availability of the labelled dataset for future use.

3.3.1 Data processing using EBS

In the progression of the research framework, the data processing phase, as illustrated in **Figure 3**, plays a critical role in translating BIM model data into actionable scheduling information. This phase bridges the gap between raw BIM data and its application in construction project management through the implementation of an EBS. The EBS is a hierarchical classification system designed to organize construction elements into a structured format [14]. It encompasses multiple levels, ranging from the project name and sub-work down to specific element categories, work packages, subprocesses, and resources.



ID	Volume	Xmin	...	Predicted _Family
341632	26.56	-9.41233	...	wall
341739	26.56	-1.41233	...	wall
346077	1.01	-9.01233	...	beam
346128	1.01	-9.01233	...	Beam
346350	0.36	-9.01233	...	Beam
346393	36.52	-9.41233	...	slab

ID	Volume	...	Predicted _Family	EBS
341632	26.56	...	wall	1.1.2.1
341739	26.56	...	wall	1.1.2.2
346077	1.01	...	beam	1.2.1.3
346128	1.01	...	beam	1.2.1.1
346350	0.36	...	beam	1.2.1.2
346393	36.52	...	slab	1.2.3.1

Figure 3 Data processing.

Within this stage, the previously extracted and minimally labelled data from the BIM model undergoes further categorization based on the predefined EBS. The labelled scheduling database, containing the ID, volume, and coordinates of each BIM element, is enhanced with predicted element families. The integration of the EBS allows for the assignment of each element to its respective work package.

The EBS assignment is automated through a custom Python script. The script classifies elements into major groups (e.g., foundation, wall, beam, slab) according to their predicted family. Each family is associated with a primary EBS code, such as "1.1" for foundation and walls, "1.2" for beams and slabs. Further, elements within each family are sorted based on their coordinates to assign a unique EBS code that reflects their sequence within the work package.

3.3.2 Work Breakdown Structure (WBS) Dictionary

Dictionaries, in the context of data structures, are characterized by their ability to map unique keys to specific values, thereby creating an organized and easily navigable database [15]. In scheduling and resource management, the WBS Dictionary acts as this type of structured database, where each element of work is a key that leads to a detailed set of attributes and values encompassing descriptions, labour requirements, and resource allocations.

Each subprocess requires specific resources to execute the tasks. In the dictionary, various resources are identified for each subprocess, such as different categories of workers. These resources are quantified in terms of hours per cubic meter (h/m³), reflecting the labour productivity rates.

3.3.3 Scheduling data base with WBS

This phase represents the confluence of BIM model data—specifically, elements identified by ID and volume—with the WBS, thereby structuring the schedule to align with the project's hierarchical breakdown. The data processing workflow incorporates a systematic approach for assigning WBS codes to each BIM element. The extracted BIM data including ID, volume, predicted family, and preliminary EBS codes, serves as the input. This data is methodically matched with the WBS Dictionary to assign the appropriate WBS codes.

The integration of WBS codes into the scheduling database is executed through an automated script, enhancing the efficiency and accuracy of the process. The script functions to parse the EBS code into its constituent parts, identifying the primary levels of the work package and checking for specific task identifiers. Matching tasks from the WBS Dictionary are integrated into the BIM data, replacing generic placeholders with specific WBS codes that correspond to distinct construction tasks.

The scheduling database with WBS, presented in CSV format, not only provides a structured view of tasks but also facilitates the mapping of resources to each task. The final output is a refined scheduling database in CSV format, which includes the WBS code assignments.

3.3.4 Scheduling data base with WBS code

Through the algorithm's implementation, each element is not only assigned a unique WBS code reflecting its project hierarchy but also tagged with specific codes for the individual subprocesses and the necessary resources associated with each subprocess. This level of specificity afforded by the unique WBS codes ensures that each element, task, and resource is precisely defined, paving the way for detailed scheduling and resource allocation. The algorithm efficiently handles multiple instances of the same element type by assigning differentiated WBS codes.

3.4 Deterministic task time calculation and resource balancing

This phase integrates a structured scheduling database with the WBS and resource productivity rates to automate and optimize the assignment of resources to tasks.

The input data, as shown in the provided dataset, encompasses a range of construction elements each with a unique ID, volume, and a predicted family classification. The dataset is further structured by WBS codes, which delineate the hierarchical placement of each element within the construction process, and by the inclusion of detailed subprocesses (Level 5) for each element (e.g., formwork, rebar, concrete pouring).

The resource allocation algorithm functions by iterating through each task, as identified by a unique WBS code, and calculating the necessary resource times based on the given productivity rates. For a given task with time and productivity rates for workers resources *A, B, C, D*, the individual resource times are calculated. The algorithm then identifies the resource that requires the minimum time to complete the task. The resource allocation proportions are calculated by normalizing the task time by the minimum resource time

The algorithm commences with the extraction of task times and productivity rates from the dataset. It employs a conditional check to ascertain the availability of productivity rates for each resource. If a rate is not provided, the corresponding resource time is not considered in the minimum time calculation. Upon calculating the minimum time and the normalized resource allocations, the algorithm appends this data to the original dataset, creating new columns for "Time (hours)" and "Q_Worker A" through "Q_Worker D," which represent the quantified allocation for each worker.

In addition to calculating resource times for individual tasks, the algorithm aggregates the total time required for all tasks within each Level 4 WBS code. This aggregation provides a comprehensive view of the time allocation for each work package considering a construction sequence. The enriched dataset, now containing both the original task details and the calculated resource allocations, is written to a new CSV file called “Scheduling data base with WBS and resources (CSV)”, ready for use in subsequent scheduling and project management activities.

3.4.1 Scheduling data base with WBS and resources

Table 1, presents a detailed view of the resource allocation for a construction project, integrating task time with productivity metrics. This table is a direct output from the previously discussed algorithmic approach, where each task within the project's scope is evaluated for its resource requirements based on productivity rates. While the time and resource allocations presented in **Table 1** have been computed deterministically, providing a solid foundation for scheduling, it is important to recognize the inherent variability in construction processes. The calculated times serve as a baseline for potential simulations to be performed with this data, which will incorporate the sequential nature of construction activities, the variability in resource productivity, and the dynamics of resource allocation.

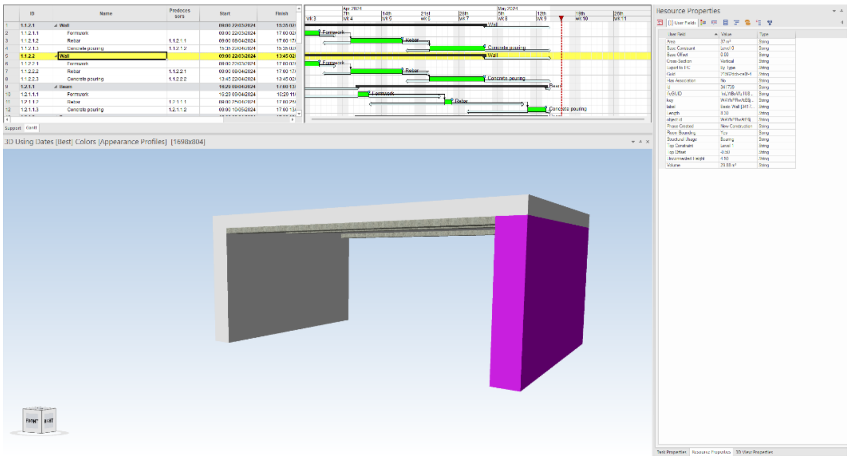
Table 1 Scheduling Database with WBS and resources (CSV)

ID	***WBS***	***Element***	***Work (m3)***	***Time (hours)***	***Worker A***	***Worker B***	***Worker C***	***Worker D***	***Time (hours)***	***Q_Worker A***	***Q_Worker B***	***Q_Worker C***	***Q_Worker D***
341632	1.1.2.1	Wall	26.56	398.4									
341632	1.1.2.1.1	Formwork	26.56		0	15	5	0	132.8		3	1	
341632	1.1.2.1.2	Rebar	26.56		0	15	0	5	132.8		3		1
341632	1.1.2.1.3	Concrete pouring	26.56		5	28	0	0	132.8	1	5.6		
341739	1.1.2.2	Wall	26.56	398.4									
341739	1.1.2.2.1	Formwork	26.56		0	15	5	0	132.8		3	1	
341739	1.1.2.2.2	Rebar	26.56		0	15	0	5	132.8		3		1
341739	1.1.2.2.3	Concrete pouring	26.56		5	28	0	0	132.8	1	5.6		
346128	1.2.1.1	Beam	1.01	15.15									
346128	1.2.1.1.1	Formwork	1.01		0	10	5	0	5.05		2	1	
346128	1.2.1.1.2	Rebar	1.01		0	10	0	5	5.05		2		1
346128	1.2.1.1.3	Concrete pouring	1.01		5	25	0	0	5.05	1	5		
346350	1.2.1.2	Beam	0.36	5.4									
346350	1.2.1.2.1	Formwork	0.36		0	10	5	0	1.8		2	1	
346350	1.2.1.2.2	Rebar	0.36		0	10	0	5	1.8		2		1
346350	1.2.1.2.3	Concrete pouring	0.36		5	25	0	0	1.8	1	5		
346077	1.2.1.3	Beam	1.01	15.15									
346077	1.2.1.3.1	Formwork	1.01		0	10	5	0	5.05		2	1	
346077	1.2.1.3.2	Rebar	1.01		0	10	0	5	5.05		2		1
346077	1.2.1.3.3	Concrete pouring	1.01		5	25	0	0	5.05	1	5		
346393	1.2.3.1	Slab	36.52	474.76									
346393	1.2.3.1.1	Formwork	36.52		0	8	4	0	146.08		2	1	
346393	1.2.3.1.2	Rebar	36.52		0	8	0	4	146.08		2		1
346393	1.2.3.1.3	Concrete pouring	36.52		5	22	0	0	182.6	1	4.4		

4.Validation trough 4D BIM simulation

In the validation phase of the automated construction schedule, a 4D BIM simulation was conducted using Synchro Pro software. The process began with the importation of a BIM model from Revit 2022, formatted in DWFX file type, along with the corresponding CSV file containing the automated construction schedule data. These files provided the necessary foundation for the simulation.

Following the import, appearance profiles were developed to visually represent various construction stages within the simulation. This step was critical in ensuring that the simulation could effectively display the progression of construction activities such as formwork, rebar placement, and concrete pouring. The core of the 4D simulation involved linking the BIM model objects to the corresponding tasks within the schedule, utilizing the Element ID as a connector as showed in figure 4. This linkage was vital for enabling a synchronized and accurate representation of the construction sequence in a temporal context.



	ID	Name	Predecessors	Start	Finish
1	1.1.2.1	Wall		09:00 22/03/2024	15:35 02
2	1.1.2.1.1	Formwork		09:00 22/03/2024	17:00 02
3	1.1.2.1.2	Rebar	1.1.2.1.1	09:00 08/04/2024	17:00 17
4	1.1.2.1.3	Concrete pouring	1.1.2.1.2	15:35 22/04/2024	15:35 02
5	1.1.2.2	Wall		09:00 22/03/2024	13:45 02
6	1.1.2.2.1	Formwork		09:00 22/03/2024	17:00 02
7	1.1.2.2.2	Rebar	1.1.2.2.1	09:00 08/04/2024	17:00 17
8	1.1.2.2.3	Concrete pouring	1.1.2.2.2	13:45 22/04/2024	13:45 02
9	1.2.1.1	Beam		16:29 09/04/2024	17:00 13
10	1.2.1.1.1	Formwork		16:29 09/04/2024	16:29 11
11	1.2.1.1.2	Rebar	1.2.1.1.1	08:00 25/04/2024	17:00 25
12	1.2.1.1.3	Concrete pouring	1.2.1.1.2	09:00 10/05/2024	17:00 13

Figure 4 BIM model and data integration in the 4D BIM software

With the BIM objects connected to the tasks, and considering the schedule data obtained from the framework the simulation generated a visual construction sequence. The real-time visualization allowed for the identification of any scheduling errors or inefficiencies.

Figure 5 represents the dynamic nature of the simulation facilitated iterative refinements to the schedule. This process of continual adjustment and enhancement ensured the automated construction schedule was not only a theoretical plan but also a practical, executable sequence.

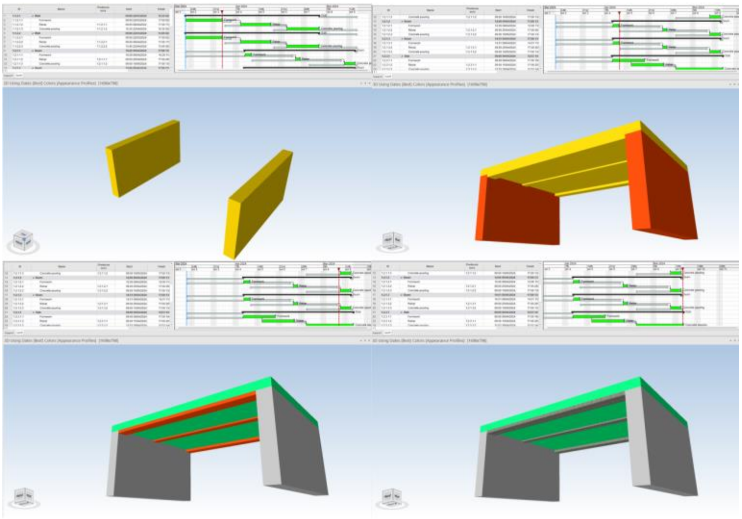


Figure *5* 4D BIM simulation

5. Conclusions

The research provided a novel framework that not only automated the extraction of data from BIM models, but also performed a data processing in an automated way providing a baseline scheduling data for conducting a schedule process.

The approach presented successfully streamlined the generation of construction schedule data, which seamlessly integrated into a 4D simulation environment. The use of Synchro Pro to validate the scheduling data affirmed the effectiveness of our methodology, utilizing the scheduling data for consuxtiong a 4d BIM simulation. The construction schedule data, output of the framework, served as a crucial baseline for conducting and refining the scheduling process, allowing for real-time validation and iterative enhancements. It demonstrated the practical application of our framework in creating foundational data for construction planning. However, the framework's application to a relatively simple bridge model does highlight certain limitations. While the automated labelling using machine learning techniques proved effective in this context, there is a possibility that it may not perform as well with more complex models. Additionally, the WBS generated by the algorithm, though it provides a structured approach to organizing project tasks and subprocesses, does not inherently represent the sequence of activities. It acts as a useful starting point but requires further refinement to capture the true sequence of construction events.

6. Future Work

While the framework capably generates resource balancing based on the minimum deterministic task times, incorporating an optimization model could further enhance resource allocation efficiency and scheduling accuracy. Testing the framework with more complex BIM models will be crucial in assessing its scalability and adaptability to diverse project demands. A critical area for future exploration will be to integrate proper sequencing into the schedule data. Developing an algorithmic solution that not only organizes tasks but also accurately sequences them according to project-specific variables will be a significant advancement for the field.

6. References

1. Koo, B., M. Fischer, and J. Kunz, *A formal identification and re-sequencing process for developing sequencing alternatives in CPM schedules.* Automation in Construction, 2007. **17**(1): p. 75-89.
2. Amer, F., H.Y. Koh, and M. Golparvar-Fard, *Automated methods and systems for construction planning and scheduling: Critical review of three decades of research.* Journal of Construction Engineering and Management, 2021. **147**(7): p. 03121002.
3. Desgagné-Lebeuf, A., et al., *Computer-assisted scheduling tools in the construction industry: A systematic literature review.* IFAC-PapersOnLine, 2019. **52**(13): p. 1843-1848.
4. Sheikhkhoshkar, M., et al., *Automated planning of concrete joint layouts with 4D-BIM.* Automation in construction, 2019. **107**: p. 102943.
5. Wang, H.-W., J.-R. Lin, and J.-P. Zhang, *Work package-based information modeling for resource-constrained scheduling of construction projects.* Automation in construction, 2020. **109**: p. 102958.
6. Faghihi, V., et al., *Automation in construction scheduling: a review of the literature.* The International Journal of Advanced Manufacturing Technology, 2015. **81**: p. 1845-1856.
7. Hong, Y., H. Xie, and I. Brilakis, *Automate Construction Scheduling at the Pre-construction Stage–A Case Study.* 2022.
8. Wu, I.-C., et al., *Bridge construction schedule generation with pattern-based construction methods and constraint-based simulation.* Advanced Engineering Informatics, 2010. **24**(4): p. 379-388.
9. Haiati, O., J. von Heyl, and S. Schmalz. *Bim and Sequence Simulation in Structural Work—Development of a Procedure for Automation*. in *Proceedings of the 24th Annual Conference of the International for Lean Construction, Boston, MA, USA*. 2016.
10. König, M., et al. *Intelligent BIM-based construction scheduling using discrete event simulation*. in *Proceedings of the 2012 Winter Simulation Conference (WSC)*. 2012. IEEE.
11. Park, J. and H. Cai, *Automatic construction schedule generation method through BIM model creation*, in *Computing in Civil Engineering 2015*. 2015. p. 620-627.
12. Nusen, P., et al., *Construction planning and scheduling of a renovation project using BIM-based multi-objective genetic algorithm.* Applied Sciences, 2021. **11**(11): p. 4716.
13. Braun, A. and A. Borrmann, *Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning.* Automation in Construction, 2019. **106**: p. 102879.
14. Sha, G., et al. *Application research on information integration technology of bridge BIM model based on 2-D EBS coding system*. in *Journal of Physics: Conference Series*. 2021. IOP Publishing.
15. Frederickson, G.N., *Implicit data structures for the dictionary problem.* Journal of the ACM (JACM), 1983. **30**(1): p. 80-94.