

Article

Exchanging Progress Information Using IFC-Based BIM for Automated Progress Monitoring

Noaman Akbar Sheik ^{1,2,*} , Peter Veelaert ¹  and Greet Deruyter ² ¹ Department of Telecommunications and Information Processing, TELIN-IPI, Ghent University, 9000 Ghent, Belgium² Department of Civil Engineering, Ghent University, 9000 Ghent, Belgium

* Correspondence: noamanakbar.sheik@ugent.be

Abstract: BIM has become an intrinsic tool in managing building projects due to its ability to comprehensively represent information in digital form. However, using BIM as an information exchange tool is still in its infancy, particularly with regard to construction progress monitoring beyond time schedule information. The current study focuses on the development of an automated progress monitoring framework based on an IFC-based BIM and provides an extensive methodology based on a structured task-based approach in accordance with the latest IFC4.x schema in four stages. The first stage creates the appropriate IFC entities, which are then enriched with their values in the second stage. The third stage integrates the actual progress information, which requires regular updating from the construction site. Finally, the fourth stage enables the retrieval of progress information, which is then reported in a user-friendly format along with the estimation of additional progress indicators. The proposed method successfully integrated the progress information into their IFC-based BIM models, demonstrating its practical use for monitoring construction progress. In the end, a web-based application was also developed that made use of progress information stored within the standardized hierarchy of the updated IFC-based BIM to facilitate efficient reporting.

Keywords: BIM updating; progress monitoring; IFC-based BIM; automation; Building Information Model; scheduling



Citation: Sheik, N.A.; Veelaert, P.; Deruyter, G. Exchanging Progress Information Using IFC-Based BIM for Automated Progress Monitoring. *Buildings* **2023**, *13*, 2390. <https://doi.org/10.3390/buildings13092390>

Academic Editor: Salvatore Antonio Biancardo

Received: 10 August 2023

Revised: 13 September 2023

Accepted: 18 September 2023

Published: 20 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Effective project management is crucial for the timely completion of building projects within schedule, budget, and quality requirements, and it necessitates constant monitoring and control of construction progress. Precise and efficient monitoring of the as-built status of construction has been widely recognized as a critical component of the building process [1–4]. It not only ensures adequate project management but also allows for early detection of deviations from the plan, providing opportunities to adopt remedial action in the early stages, which can save time and money [5–7]. Hence, accurate and efficient tracking of the as-built status of buildings under construction is essential for successful project control. To address this need, Building Information Modeling (BIM) has emerged as a valuable tool for construction project management due to its capability of facilitating communication and collaboration among project stakeholders [8–11].

BIM can provide a comprehensive and accurate representation of the building, including its geometry, spatial relationships, and non-graphic information such as cost and schedule data. BIM has been extensively used in the design and planning phases of construction projects, leading to improved accuracy, reduced errors, and increased efficiency. The construction industry has undergone significant changes with the adoption of Building Information Modeling (BIM) technology, which has enabled effective project management through enhanced collaboration, communication, and project control [2,12–14]. However, despite the benefits of BIM, its full potential has yet to be realized in construction progress

monitoring and reporting [15]. A major challenge in this regard is the effective exchange and management of information throughout the entire construction process.

As a result, there is a need to explore the use of BIM for communicating progress information during construction monitoring to unlock its potential for effective project management. This study aims to investigate the use of Industry Foundation Classes (IFC), an open standard for BIM data exchange, as a means of effectively communicating progress information during construction progress monitoring. The IFC is a widely recognized open and neutral file format for standardized BIM data that is acknowledged by prominent international standardization organizations, including the International Organization for Standardization (ISO), the European Committee for Standardization (CEN), and the German Institute for Standardization (DIN) [16–19]. The IFC standard, maintained by the non-profit organization BuildingSMART [20], facilitates the interoperability of BIM software applications (<https://www.accasoftware.com/en/bim-software>). It is described by a general schema, the IFC schema, utilized to formally describe the semantic information of buildings and infrastructure [21]. The implementation of IFC-based BIM models requires the extraction of relevant information through the IFC schema [3]. Similarly, updating the model may involve creating, modifying, or updating IFC entities. However, this process can be challenging to perform, especially when considering the complexity of the data and the specific goals required.

Recent studies have highlighted the potential of IFC-based BIM for various purposes [22–32] including the domain of construction monitoring [2,3,5,33–35]. However, there is a gap in the research when it comes to utilizing IFC-based BIM for updating progress information beyond just schedule updates. Furthermore, there is a need for a comprehensive approach to exchanging progress information using IFC-based BIM, leveraging the latest IFC4 schema.

In comparison to the outdated IFC2x3 schema, the IFC4.x [36] schema represents a significant advancement in BIM technology due to several improvements and new features; hence, it is a more suitable schema for progress monitoring. One major difference is that IFC4.x now uses the ISO 8601 string format for date and time definitions, reducing the model footprint. In addition, time-related information such as schedule or actual Start/finish/duration, completion, status time, etc. are stored in the 'IfcTaskTime' entity in IFC4.x, while in IFC2x3, they were stored in 'IfcScheduleTimeControl'. Furthermore, the definitions for construction resources have been reworked and now include the notion of resource types, which has improved the assignment of resources to schedules and costs. Similarly, the concept of a process type and relevant subtypes has been introduced, allowing for the sequencing of tasks. The new process and cost definitions in IFC4.x significantly reduce the model footprint when capturing the geometrical details along with time, cost, and other information, making it a more effective schema for progress monitoring in BIM projects.

The research focuses on the development of an automated progress monitoring framework that adopts a structured task-based approach and utilizes the IFC-based BIM to exchange progress information (independent of data collection technique), including time, cost, and other additional progress parameters for construction progress monitoring. This framework leverages the latest IFC4 data to ensure standardized and consistent progress monitoring without any software dependency, providing project stakeholders with up-to-date and accurate insights into the project's status. The methodology presented in this study involves four main steps: (1) integration of progress-related entities with the IFC-BIM; (2) inputting planned parameters according to progress information obtained from construction; (3) updating actual progress into the IFC-based BIM model; and (4) reporting progress information directly from the updated BIM. By utilizing this methodology, we aim to enhance the accuracy and efficiency of progress monitoring and control in construction projects. The updated IFC-based BIM model reflects both as-planned and as-built progress information, which provides a comprehensive and up-to-date view of the project status. Additionally, the methodology allows for the reporting of progress information

with additional progress parameters, such as earned value, estimated completion time, and inspection comments, providing a detailed and holistic understanding of the project progress and facilitating effective project tracking.

The paper is organized as follows: Section 2 provides a literature review of the existing research on construction progress monitoring using IFC-based BIM, focusing on the methodologies used and the progress parameters monitored. Section 3 details the proposed methodology for automated progress monitoring using IFC-based BIM, describing the four main steps involved in the process. Section 4 presents the results and related discussion of the implementation of methodology in IFC-based models. Finally, Section 5 concludes the paper by summarizing the key contributions of the study and highlighting its significance for the field of construction progress monitoring.

2. Literature Review

Efficient progress monitoring and control are essential components of successful construction project management [4,6,37–40]. The significance of accurate monitoring has been underscored by numerous studies in the field of construction management, highlighting its pivotal role in ensuring project completion within specified timelines and budgets while enabling timely identification of any potential issues that may arise during the building process [3,5,41–43].

Efficient construction progress monitoring requires up-to-date and accurate as-built progress information. However, the traditional methods for collecting and processing this information are time-consuming and labor-intensive, leading to the possibility of missing or inaccurate information [42,44–48]. Although automated progress monitoring systems are being developed with time to address this issue, there is still a significant challenge in performing the exchange of progress information in terms of automation, accuracy, standardization, and interoperability [33]. This emphasizes the necessity for a proficient solution that allows the automated exchange of progress information from a construction site into a centralized digital platform, such as BIM. The current research exploits the use of BIM to facilitate information exchange for automated construction progress monitoring.

To achieve this goal, it is important to understand the potential benefits of BIM in the construction industry. BIM is positioned at the forefront of digitization initiatives in the construction industry and is becoming progressively integrated with other digital technologies as part of the broader Industry 4.0 framework [49–51]. Studies have shown that Building Information Modeling (BIM) has transformative impacts on the design, construction, and operation of buildings, serving as a valuable repository of information to enhance these processes [8,52–54]. Moreover, digital planning and construction supported by BIM have the potential to yield significant cost reductions of 13% to 21%, while the operation phase may see cost reductions ranging from 10% to 17% [19,55].

Additionally, BIMs integration with other technologies presents an opportunity to revolutionize infrastructure construction and maintenance [56]. Consequently, BIM has gained popularity in construction projects, with several countries mandating its use in large projects. In Europe, there has been a substantial increase in BIM applications from 2012 to 2017 [57,58]. Therefore, it is essential to explore the potential of integrating BIM with automated progress monitoring systems to optimize the construction progress monitoring process, which can lead to better project outcomes and increased efficiency.

Building Information Modeling (BIM) is a digital representation of a construction project, providing comprehensive information about its physical and functional characteristics. The information content of BIM is diverse and tailored to specific project requirements, implying that it is not standardized. BIM encompasses both 3D geometry and semantics, where the former provides the basis for consistent technical drawings and enables clash detection [59] and collision resolution during the design phase [60]. On the other hand, semantics complements the meaning of objects by providing alphanumeric data, including mechanical and thermal properties, materials, associations, aggregation, and other relationships between objects in the model [61]. Consequently, various analyses, simulations, and

advanced controls can be performed on the BIM model, facilitating effective collaboration, coordination, and decision-making throughout the project lifecycle [52].

In the realm of construction management, 4D BIM (BIM with time-related schedule information) has improved project management efficiency. It has been utilized for progress monitoring through comparison with 3D scan models [43,62,63]. Other studies have demonstrated its use in quality monitoring and project scheduling improvements by combining it with quality information models and tabu-search algorithms, respectively [64,65]. Although 4D BIM offers benefits for on-site construction management, it also poses challenges such as labor-intensive and time-consuming updating of models to reflect as-built conditions during construction [33,66] and difficulties in interoperability between stakeholders, phases, and BIM software [67,68]. Additionally, 4D BIM models rely on specific software, which can make information inaccessible once separated from the software, leading to inefficiencies in managing buildings and creating a dependence on specific tool vendors [69,70].

To address these challenges, it is vital to establish standardized BIM content with comprehensive guidelines and standards to govern BIM content across the construction industry. However, the successful adoption of BIM relies heavily on integrating mono-discipline BIM models into a cohesive, multi-disciplinary model. This requires harmonization and interoperable information exchange between individual models created using specialized BIM software tools to achieve a universal and neutral data format [52,71]. The IFC data format, globally recognized as the standard for BIM data exchange and sharing, provides a solution for this challenge.

IFC is an internationally accepted standard for data sharing in construction and facility management. It enables the exchange and sharing of BIM data between various software applications used in construction and facility management for both buildings and infrastructure. The IFC standard is the basis for a universal and consistent file format for information exchange and data sharing. It is an open and non-proprietary schema based on the EXPRESS language for data specifications, ensuring a consistent approach to data transfer. Hence, certified BIM software applications can convert their BIM models into the neutral IFC model, facilitating interoperability and collaboration among different stakeholders in the construction industry [16,52]. Despite some drawbacks, such as the incomplete association of modeled elements with semantics and potential loss of information during export to the IFC format [42,72,73], the IFC-based BIM schema continues to be enhanced and is widely employed in contemporary construction projects. The IFC standard has been widely utilized and studied in various areas of the construction industry, such as data standards for facilities management classes [22], energy consumption [23,24], virtual construction [25,26], and project performance evaluation [27]. In addition, several scholars have extended the IFC standard for construction management information, including the integration of completion and design information [28], plan information storage [29], quantity take-off [30,31], GIS integration [74], and daily 4D BIM visualization [32].

In the field of construction progress monitoring, Golparvar-Fard, Pena-Mora and Savarese [2] and Son, Kim and Kwon Cho [35] employed the IFC-based BIM, with manually integrated time-related schedule information, for progress detection by comparing it with reconstructed as-built point cloud obtained from photographs and laser scans, respectively. Similarly, there are some studies [3,5] that performed the registration of an as-built model with the IFC-based BIM model where geometrical details were extracted from the latter to obtain matching features for alignment. Although these methods utilized the IFC-based BIM as the as-planned model, they did not address the exchange of progress information using IFC-based BIM models. In addition, Hamledari, McCabe, Davari and Shahi [33] developed an automated method using the IFC-based BIM for updating time-related schedule information. This method involved adjusting the task-object relationship based on the level of progress details, updating actual progress entities according to their completion ratios, and estimating the non-completed planned entities based on actual progress. Although this method automates the updating process, it is limited in its scope as

it only focuses on time-related schedule information and does not incorporate cost or other crucial factors for effective progress monitoring. Furthermore, the method was based on the IFC2x3 schema, which may not be compatible with the latest IFC4.x schema that natively supports progress information through different entities and approaches. Therefore, there is a research gap in developing a comprehensive and integrated approach that leverages the full potential of BIM as an information exchange tool for construction progress monitoring by incorporating different types of progress parameters such as time, cost, or any additional information using the latest IFC4.x schema.

The current study aims to address this gap and explores the application of IFC-based BIM models for construction progress monitoring. The research article proposes a methodology for updating the model with current progress information, including scheduling, cost, and other key performance indicators, to improve the accuracy and effectiveness of project monitoring. The methodology enables not only the updating of progress information but also the addition of semantic information such as comments, notes, and delay reasons. This study culminates by demonstrating the visualization and reporting of progress information from the updated IFC-based BIM model through the use of color-coded 3D models, graphs, and critical information, providing a comprehensive solution for information exchange to facilitate effective construction project monitoring.

3. Methodology

The proposed methodology presents a systematic approach to enable the effective communication of progress information between IFC-based BIM and construction sites for efficient and accurate construction progress monitoring. The approach comprises four stages, as shown in Figure 1. The first stage involves the integration of progress-related entities with the IFC-BIM according to progress information obtained from construction. In the second stage, the planned parameters are inputted into the relevant entities of the IFC-based BIM. The third stage allows the updating of actual progress into the IFC-based BIM model. Finally, the last stage details the reporting of progress information directly from the updated BIM, providing stakeholders with up-to-date and accurate insights into the project's status.

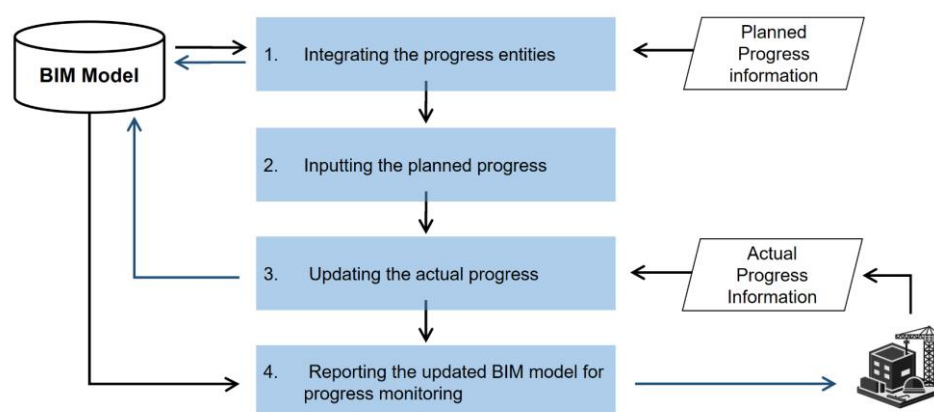


Figure 1. Flowchart of proposed methodology.

The proposed methodology requires the input of planned and actual progress from the construction site. The planned progress comprises the IFC-based BIM model, the planned progress schedule, and the budgeted cost schedule. Similarly, the actual information includes the actual progress, actual cost information, and comments related to individual building components obtained from the construction site.

The construction of a building includes various physical components such as walls, doors, slabs, etc. The overall completion of the building is reliant on the individual progress of each of these components, and any delays or cost overruns can be traced back to their performance. These building components are constructed through a set

of different work activities (or tasks), and the completion status of the component is signified by the advancement of these tasks. For example, the construction of a wall may involve tasks such as layout, bricklaying, and plastering, which together determine the completion status of the wall. An effective and accurate approach to monitoring construction progress requires a framework that is centered around the individual tasks of the building components. The proposed methodology adopted a task-based approach for the exchange of progress information with IFC-based BIM, where the progress entities for tasks, building components, and the overall project are designed accordingly in all stages of the method.

IFC schema for progress monitoring.

The proposed methodology leverages the IFC schema for exchanging progress information during construction monitoring. This IFC schema [75] is an object-oriented data model that organizes concepts into four layers: core, interoperability, domain, and resource. The core layer includes the most general classes, such as *IfcRoot*, *IfcRelationship*, and *IfcObject*, which are used to define objects, relationships, and properties. The interoperability layer includes classes that specialize in the product extension schema, providing more detailed information. The domain-specific layer specifies classes for particular domains, while the resource layer further describes objects at other levels. A key feature of the schema is inheritance between classes, with the *IfcRoot* class serving as the initial root class from which other classes are derived in three directions: physical and conceptual objects (*IfcObjectDefinition*), relations between objects (*IfcRelationships*), and object properties (*IfcPropertyDefinition*). The schema is organized in hierarchy, with layers grouped according to function and specialization.

An IFC entity consists of attributes and their respective values. The attributes represent the type of information, while the values provide specific details about the object being represented. This allows for a standardized and structured way of representing building information throughout the construction process, making it easier to exchange and interpret data between different software applications and stakeholders. The proposed method is in line with the IFC4.3 schema, and the IFC entities used in the method are shown in Figure 2 with their hierarchy.

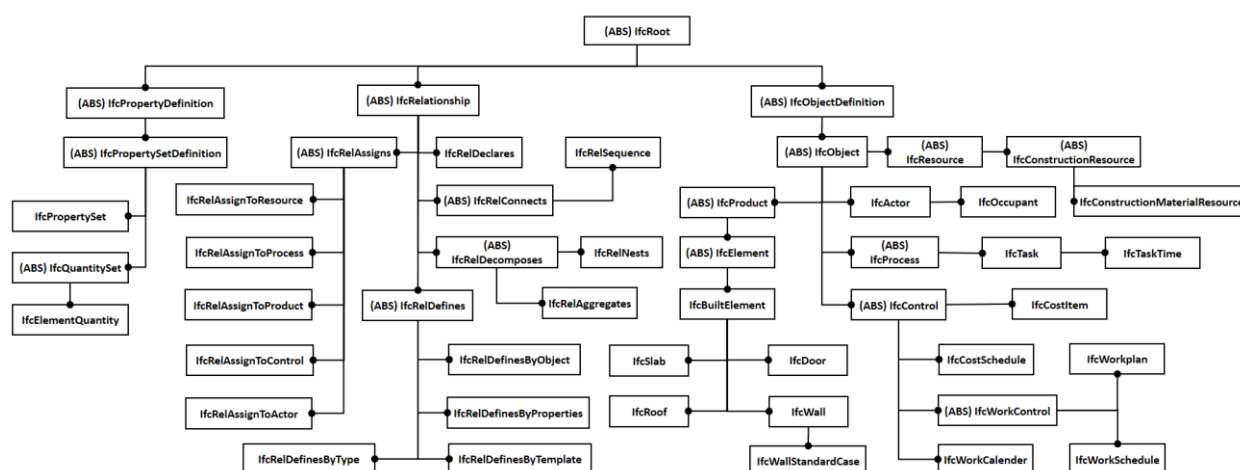


Figure 2. IFC entities related progress information in hierarchy.

In the IFC-based BIM model, building components such as walls, slabs, columns, and doors can be represented by corresponding subtype IFC entities of 'IfcBuiltElement', such as 'IfcWall', 'IfcSlab', 'IfcColumn', and 'IfcDoor', respectively. The work tasks involved in the construction of these components are represented by the 'IfcTask' entity. Furthermore, a suitable relationship entity such as 'IfcRelAssignsToProduct' is used to express their association between entities. In cases where there are multiple activities or tasks, additional 'IfcTask' entities can be assigned and linked to the main task entity through

a nest relationship ‘IfcRelNests’ entity. Furthermore, property entities can also be linked to store additional information. An example of the IFC representation of tasks with their defined sequences involved in the construction of the wall is illustrated in Figure 3.

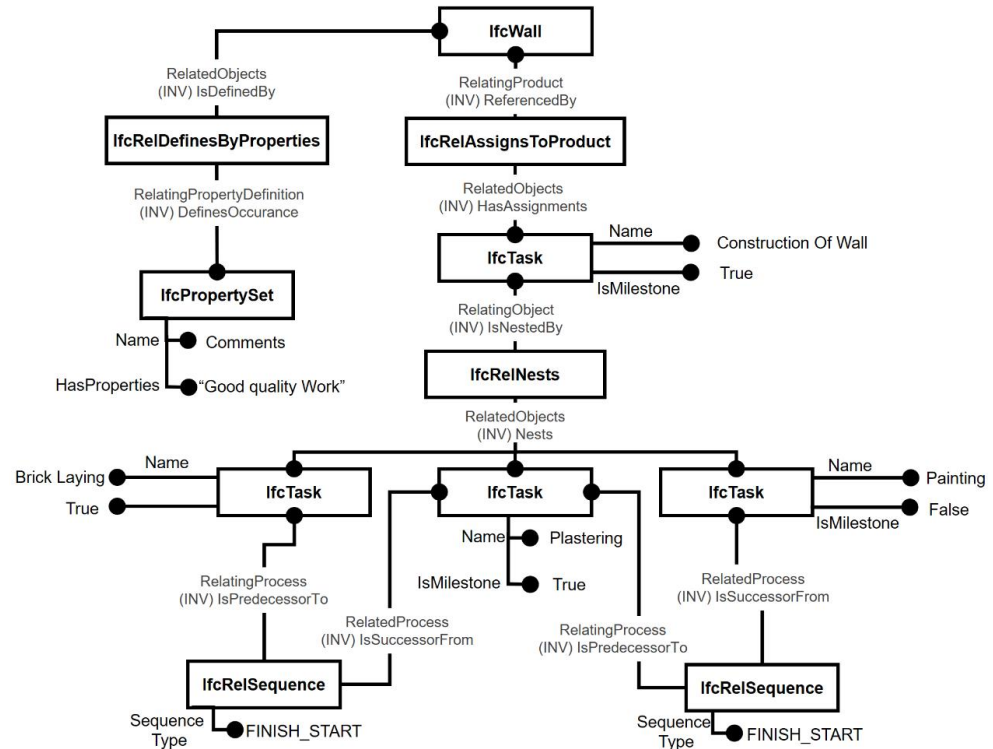


Figure 3. An example of entities associated with a building component in the IFC-based BIM model.

The utilization of IFC-based BIM not only enables the updating of the progress entities related to schedule, cost, and other indicators but also other semantic information such as comments, important notes, delay reasons, etc. Furthermore, it is not necessary for the planned information, such as the planned schedule or cost schedule, to be integrated into the IFC-based BIM model beforehand, as the proposed method deals with this challenge (stages 1 and 2) in an automated way. Similarly, the proposed method not only updates the IFC-based BIM with progress information (stages 2 and 3) but also enables the retrieval of progress information in the form of progress indicators for construction progress monitoring in the last stage.

3.1. Integrating the Progress Entities into IFC-Based BIM

The first stage of the proposed methodology involves upgrading the IFC-based BIM by integrating additional IFC entities to accommodate progress information related to time, cost, and other relevant data. The original BIM model may lack the necessary progress entities; hence, the model is adapted by automating the creation process of standardized progress entities using the IFC4 schema. The resulting BIM model incorporates these IFC entities to comprehensively represent planned and actual progress information.

In this stage, the IFC-based BIM and planned progress information are processed to output a revised BIM model that includes progress-related entities. The aim is to minimize the overall size of the output IFC BIM and prevent an increase in the complexity of IFC entities. To achieve this goal, the process is formulated to create only the necessary IFC entities for adjusting the maximum progress information into the suitable IFC4 entities without any duplication.

To include any progress information in BIM, the appropriate IFC entities are created along with their linkage with other entities within an IFC-based BIM. A verification step has been implemented to establish that the appropriate IFC entity, preferably an IFC4

native entity, is opted for specific progress information while preventing the duplicate creation of that entity. This step determines the presence of the entity responsible for storing the specific information in the input BIM model. In the event that the entity is absent, the process of its creation is initiated that utilizes the native IFC4-based entities, as demonstrated in Figure 4.

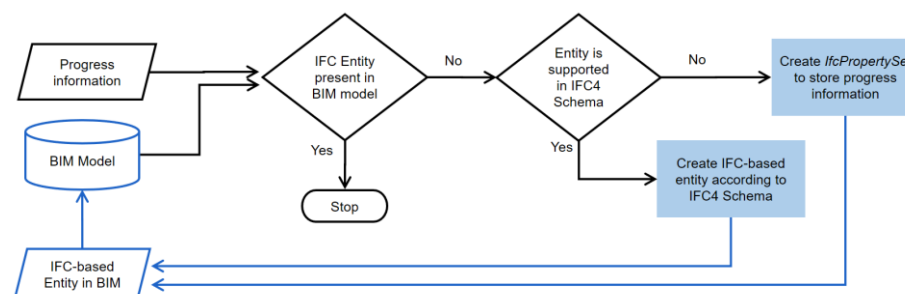


Figure 4. Flowchart of verification step to create an IFC entity based on progress information.

The proposed method, through this approach, facilitates the creation of progress entities within any IFC-based BIM, regardless of the presence of an integrated time or cost schedule. Additionally, the method remains effective irrespective of whether the IFC-based BIM model is being processed for the first time or not. For example, in the case when the same BIM model is again processed for updating or retrieval of progress information, the proposed method will skip this creation stage as all the required entities were already created during the first processing. In this way, the creation of progress entities is standardized and efficient while ensuring compliance with the IFC4 schema at all stages.

The proposed method utilizes the IFC4 schema to store progress information in the IFC model by creating appropriate IFC entities and linking them with other organized entities in the hierarchical structure. Each IFC entity has a defined number of IFC attributes and may have additional IFC properties. The IFC attributes, with their fixed names defined by buildingSMART, play an important role in the identification of the entities. The IFC entities inherit attributes from their supertype or parent entities in the hierarchy (demonstrated in Figure 5), in addition to their directly attached entity attributes. This hierarchical organization of information ensures consistency while maintaining data integrity in the IFC model. By following this standardized structure, the creation and retrieval of progress entities can be streamlined and automated, making the process more efficient and accurate.

Figure 5 highlights the hierarchy of some object classes. The object classes *IfcSlab* and *IfcWall* are the subtypes of *IfcBuiltElement* and thus share the common attributes (highlighted in green and blue) between them, with the exception of their direct attributes. In contrast, *IfcTask* also shares some attributes (highlighted in green) with both *IfcSlab* and *IfcWall*; however, only up to the *IfcObject* level. As a result, *IfcTask* has the same initial attributes as the other two classes, while the remaining attributes differ based on their position in the hierarchical structure.

To add new information to the IFC-based BIM, it is necessary to create a new entity with the appropriate IFC subtype. This defines the set of attributes and properties applicable to the entity. The entity can be linked to other entities using required relationship classes, such as 'IfcRelAssignsToControl', allowing the establishment of relationships between entities and ultimately enabling the organization of information using the IFC schema. Compliance with the IFC schema is critical while creating new entities to ensure that the correct attributes and properties are included and the relationship with other entities is established correctly, which in turn facilitates the organization of information using the IFC schema.

IfcSlab		IfcWall		IfcTask	
IfcRoot	1 GlobalId	IfcGloballyUniqueId			
	2 OwnerHistory	IfcOwnerHistory			
	3 Name	IfcLabel			
	4 Description	IfcText			
IfcObjectDefinition	HasAssignments	IfcRelAssigns			
	Nests	IfcRelNests			
	IsNestedBy	IfcRelNests			
	HasContext	IfcRelDeclares			
	IsDecomposedBy	IfcRelAggregates			
	Decomposes	IfcRelAggregates			
IfcObject	HasAssociations	IfcRelAssociates			
	5 ObjectType	IfcLabel			
	IsDeclaredBy	IfcRelDefinesByObject			
	Declares	IfcRelDefinesByObject			
	IsTypedBy	IfcRelDefinesByType			
IfcProduct	IsDefinedBy	IfcRelDefinesByProperties			
	6 ObjectPlacement	IfcObjectPlacement			
	7 Representation	IfcProductRepresentation			
	ReferencedBy	IfcRelAssignsToProduct			
IfcProcess	PositionedRelativeTo	IfcRelPositions			
	ReferencedInStructures	IfcRelReferencedInSpatialStructure			
	6 Identification	IfcIdentifier			
	7 LongDescription	IfcText			
IfcElement	IsPredecessorTo	IfcRelSequence			
	IsSuccessorFrom	IfcRelSequence			
	OperatesOn	IfcRelAssignsToProcess			
	8 Tag	IfcIdentifier			
	FillsVoids	IfcRelFillsElement			
	ConnectedTo	IfcRelConnectsElements			
	IsInterferedByElements	IfcRelInterferesElements			
	InterferesElements	IfcRelInterferesElements			
	HasProjections	IfcRelProjectsElement			
	HasOpenings	IfcRelVoidsElement			
	IsConnectionRealization	IfcRelConnectsWithRealizingElements			
	ProvidesBoundaries	IfcRelSpaceBoundary			
IfcTask	ConnectedFrom	IfcRelConnectsElements			
	ContainedInStructure	IfcRelContainedInSpatialStructure			
	HasCoverings	IfcRelCoversBldgElements			
	8 Status	IfcLabel			
IfcBuiltElement (Nil)					
IfcSlab					
IfcTask	9 PredefinedType	IfcSlabTypeEnum			
	9 PredefinedType	IfcWallTypeEnum			

Figure 5. IFC entities IfcSlab, IfcWall, IfcTask having common attributes based on their hierarchical structure.

The IFC-based entities in the proposed method are always created with IfcRoot attributes, such as GlobalId, Name, and Description. The GlobalId attribute is populated with a 22-character string based on the IfcGloballyUniqueId type. The OwnerHistory attribute records the identification details of the person responsible for creating the entities and is populated using the IfcOwnerHistory entity. The Name and Description attributes are filled with relevant information pertaining to the entity.

The process of integrating progress information into the IFC-based BIM requires a specific approach to ensure that the relevant entities are created and linked appropriately. This involves creating separate IFC entities for time, cost, and other information and establishing the necessary connections between them. The detailed procedures outlined below for each type of information are essential for ensuring that the resulting BIM model comprehensively represents progress information.

3.1.1. Time-Related Entities

The IFC4 schema already supports scheduling information for individual building components. Hence, the proposed method creates appropriately supported entities to store the relevant information as per IFC standards. The inclusion of time-related information into BIM includes creating native entities such as *IfcTask*, *IfcTaskTime*, *IfcWorkPlan*, *IfcWorkSchedule*, *IfcWorkCalendar*, etc. for each building component 'IfcBuiltElement' along with their respective relationships, as shown in Figure 6.

The proposed method is a task-based approach; hence, the entities are created accordingly. The Algorithm 1, as shown below, outlines the steps to create time-related IFC entities in a Building Information Model (BIM) to include information on planned and actual scheduling parameters. It inputs the IFC-based BIM and the planned work schedule to output an updated BIM with the necessary schedule entities created in accordance with the work activities or tasks. The algorithm creates entities for (*i*-th) work main tasks and their (*n*-th) sub-tasks under each building component, establishes their sequential relationship,

and links them with relevant time-related schedule entities such as task time, work plan, schedule, and calendar.

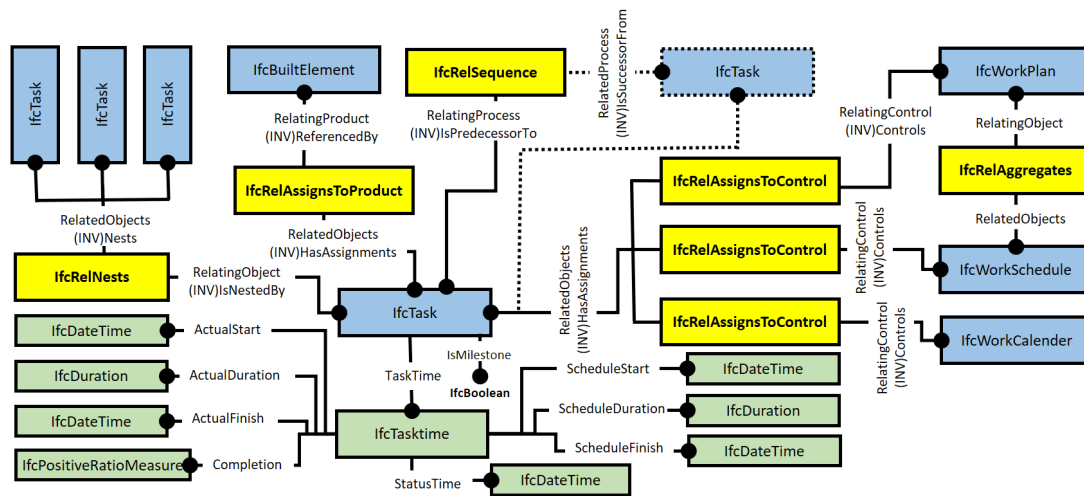


Figure 6. Structure of IFC-based entities created to include time-related progress information.

Algorithm 1: Creating time-related IFC entities into IFC-based BIM model

Input: IFC-based BIM, Planned Time-related Schedule

Output: Updated IFC-based BIM

```

1  for each 'IfcBuiltElement  $i$ ' in IFC-based BIM do
2      create an 'IfcTask  $i$ ' entity
3      create an 'IfcTaskTime  $i$ ' entity
4      link 'IfcTask  $i$ ' with 'IfcTaskTime  $i$ ' as TaskTime
5      create relationship 'IfcRelAssignsToProduct' to assign 'IfcTask  $i$ ' with 'IfcBuiltElement  $i$ '
6      if sub-activities exist then
7          create relationship 'IfcRelNests' to assign 'IfcTask  $i$ ' as RelatingObject
8          for each sub-activity do
9              create 'IfcTask  $n$ ' entity
10             create 'IfcTaskTime  $n$ ' entity
11             link 'IfcTask  $n$ ' with 'IfcTaskTime  $n$ ' as TaskTime
12             link 'IfcRelNests' with 'IfcTask  $n$ ' as RelatedObjects
13         end for
14     end if
15 end for
16 create relationship 'IfcRelSequence' to establish sequential relationship between all 'IfcTask' entities
17 create 'IfcWorkSchedule'
18 create relationship 'IfcRelAssignsToControl' to assign 'IfcWorkSchedule' with all 'IfcTask' entities
19 create 'IfcWorkPlan'
20 create relationship 'IfcRelAssignsToControl' to assign 'IfcWorkPlan' with all 'IfcTask' entities
21 create relationship 'IfcRelAggregates' to assign 'IfcWorkPlan' with 'IfcWorkSchedule'
22 create 'IfcWorkCalender'
23 create relationship 'IfcRelAssignsToControl' to assign 'IfcWorkCalender' with all 'IfcTask' entities

```

The workflow of the algorithm is briefly explained below.

The algorithm starts by looping through each '*IfcBuiltElement*' in the IFC-based Building Information Model (BIM). For each *i*-th building component represented by the '*IfcBuiltElement*' entity, the algorithm creates an '*IfcTask*' entity that represents a specific piece of work that is to be accomplished. The attribute '*IsMilestone*' of '*IfcTask*' represents whether the particular task is critical for completion; it is not optional, and it is initially set to 'False' with the intention to revise it at the final updating stage. To associate the time-relation information with the task, a new entity '*IfcTaskTime*' is created and linked to the '*IfcTask*' through the *TaskTime* attribute such that *IfcTask.Tasktime* = *IfcTaskTime*. Through this linkage, the time-related information of tasks, including their start and end dates as well as duration, is stored in the attributes of '*IfcTaskTime*' that are illustrated in Figure 6. The '*IfcRelAssignsToProduct*' relationship entity is then created to assign the '*IfcTask*' to the '*IfcBuiltElement*' through the attributes in such a way that *IfcRelAssignsToProduct.RelatingProduct* = *IfcBuiltElement* and *IfcTask.HasAssignments* = *IfcRelAssignsToProduct*.

Next, if there are multiple sub-tasks present under the main task in planned information, then the already created '*IfcTask*' acts as the main task, and the sub-tasks are created separately. The algorithm creates another loop to go through each sub-task. For each *n*-th sub-task, the algorithm creates a new '*IfcTask*' and '*IfcTaskTime*' entity and links the '*IfcTaskTime*' to the '*IfcTask*'. The algorithm then creates the '*IfcRelNests*' relationship entity to assign the sub-task to the main task of the building component.

After creating all the entities for '*IfcBuiltElement*', the algorithm creates the '*IfcRelSequence*' relationship entity to establish the sequential relationship between '*IfcTask*' entities according to planned tasks using the attributes *RelatingProcess* and *RelatedProcess* that represent the predecessor and successor tasks, respectively. To manage the work schedule, the algorithm creates an '*IfcWorkSchedule*' entity and assigns it to all '*IfcTask*' using the '*IfcRelAssignsToControl*' relationship entity, linking the entities through relationship *IfcRelAssignsToControl.RelatingControl* = *IfcWorkSchedule*. Similarly, an '*IfcWorkPlan*' entity is then created to represent the detailed plan of the work schedule, assigns it to all '*IfcTask*' using the '*IfcRelAssignsToControl*' relationship entity, and aggregates it to the '*IfcWorkSchedule*' using the '*IfcRelAggregates*' relationship entity. Finally, the algorithm concludes by creating an '*IfcWorkCalendar*' entity to represent the working days and hours and assigning it to all '*IfcTask*' entities in the same way, using the '*IfcRelAssignsToControl*' relationship entity.

The algorithm creates the time-related information for tasks (under the building component) and sub-tasks (under the main tasks) with appropriate entities in a standardized form. However, the project-based time-related entities are not created as they already exist in IFC-based BIM through a project-related *IfcPropertySet* (*Pset_ConstructionResource*) and are updated based on tasks and subtasks in the later stage.

3.1.2. Cost-Related Entities

The proposed method creates cost-related entities in the form of resources and cost items. The resource entities, such as '*IfcConstructionMaterialResource*' are assigned for the cost of tasks, while the '*IfcCostItem*' entities are employed to represent the costs of building components and the overall project. Figure 7 illustrates the structure of IFC entities being created to represent the cost parameters for construction progress monitoring.

Algorithm 2, as shown below, creates the appropriate cost-related entities along with the corresponding relationships to retain the clearly defined cost information of each work task, building component, and overall project in a standardized form. The algorithm is processed in two phases to create IFC entities, both for planned and actual progress, respectively. In each phase, the '*PredefinedType*' attribute of the '*IfcCostSchedule*' entity is set to 'Budget' and 'USERDEFINED' to specify the cost of planned or actual work, respectively. Hence, two entities, indicating the cost associated with planned and actual work, will be created in the end for each task or building component, as shown in Figure 7,

where the left and right sides, respectively, represent the planned and actual entities created to represent the cost. Additionally, the program assumes that the input IFC-based BIM has already been processed with the previous algorithm (Algorithm 1) and therefore contains the relevant 'IfcTask' entities that will be further utilized in this algorithm.

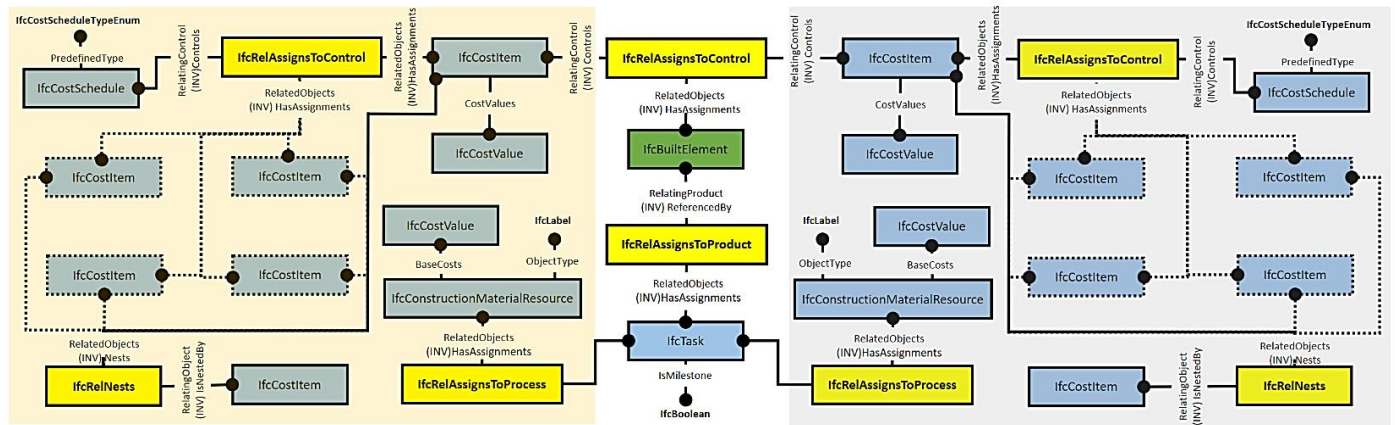


Figure 7. The Structure of IFC-based entities was created to include cost-related progress information. Entities on the left and right sides representing the planned and actual cost information.

Algorithm 2: Creating cost-related entities into IFC-based BIM model

Input: IFC-based BIM, Planned Cost-Related Schedule

Output: Updated IFC-based BIM

```

1  create an 'IfcCostSchedule' entity
2  create an 'IfcRelAssignsToControl' entity
3  link 'IfcRelAssignsToControl' with 'IfcCostSchedule' as RelatingControl
4  create an 'IfcCostValuei' entity
5  create an 'IfcPhysicalQuantityi' entity
6  create an 'IfcCostItemi' entity
7  link 'IfcCostItemi' entity with 'IfcCostValuei' and 'IfcPhysicalQuantityi' entity
8  create a relationship 'IfcRelNests' entity to assign 'IfcCostItemi' as RelatingObject
9  for each 'IfcBuiltElementi' in IFC-based BIM do
10   create an 'IfcCostValuei' entity
11   create an 'IfcPhysicalQuantityi' entity
12   create an 'IfcCostItemi' entity
13   link 'IfcCostItemi' entity with 'IfcCostValuei' and 'IfcPhysicalQuantityi' entity
14   create a relationship 'IfcRelAssignsToControl' entity to assign 'IfcCostItemi' with 'IfcBuiltElementi'
15   link 'IfcRelAssignsToControl' with 'IfcCostItemi' as RelatedObjects
16   link 'IfcRelNests' entity with 'IfcCostItemi' as RelatedObjects
17   if sub-activities exist then
18     for each sub-activity do
19       create an 'IfcCostValuen' entity
20       create an 'IfcPhysicalQuantityn' entity
21       create an 'IfcConstructionMaterialResourcen' entity
22       link 'IfcConstructionMaterialResourcen' with 'IfcCostValuen' and 'IfcPhysicalQuantityn' entity
23       create a relationship 'IfcRelAssignsToProcessn' entity to assign 'IfcTaskn' with 'IfcConstructionMaterialResourcen'
24   else
25     create an 'IfcCostValuei' entity
26     create an 'IfcPhysicalQuantityi' entity
27     create an 'IfcConstructionMaterialResourcei' entity
28     link 'IfcConstructionMaterialResourcei' with 'IfcCostValuei' and 'IfcPhysicalQuantityi' entity
29     create a relationship 'IfcRelAssignsToProcessi' entity to assign 'IfcTaski' with 'IfcConstructionMaterialResourcei'
30  end for
  
```

The details of this algorithm are as follows.

The algorithm begins by creating an 'IfcCostSchedule' entity with a value of a 'Pre-defined Type' attribute, which has the type of 'IfcCostScheduleTypeEnum', set at 'Budget' or 'USERDEFINED' corresponding to the planned or actual type of information, respectively.

Subsequently, the relationship entity 'IfcRelAssignsToControl' is established for the connection of the 'IfcCostSchedule' entity (IfcRelAssignsToControl.RelatingControl = IfcRelAssignsToControl) with the cost item entities that will be created for each building component. Next, the algorithm creates an 'IfcCostItem' entity and links it with the 'IfcCostValue' entity to represent the total cost of the building. Later, an 'IfcCostItem' entity along with 'IfcCostValue' are created to represent the cost information for each 'IfcBuiltElement'. This 'IfcCostItem' is then assigned to 'IfcBuiltElement' using the relationship 'IfcRelAssignsToControl' entity such that IfcRelAssignsToControl.RelatingControl = IfcCostItem and IfcBuiltElement.HasAssignments = IfcRelAssignsToControl.

Later in the algorithm, for each work activity, the 'IfcTask' entity linked to the 'IfcBuiltElement', resource entities are created. The process begins by creating the 'IfcConstructionMaterialResource' entity along with the linked 'IfcCostValue' and 'IfcPhysicalQuantity' entities. This entity is then linked with 'IfcTask' using the relationship 'IfcRelAssignsToProcess' entity, where IfcRelAssignsToProcess.RelatingProcess = IfcTask and IfcConstructionMaterialResource.HasAssignments = IfcRelAssignsToProcess.

Finally, all the individual 'IfcCostItem_i' entities of building components are nested with the 'IfcCostItem_i' to ensure their linkage for the representation of the total building cost. Similarly, these entities are linked with the initially created 'IfcCostSchedule' using the relationship 'IfcRelAssignsToControl'. In this way, all the required cost-related entities are created either for cost, planned, or actual progress.

3.1.3. Additional Information

During construction progress monitoring, it is common to collect additional information that may not fit into standard IFC entities. This information, such as progress-related comments and non-standard progress parameters, can be stored in customized 'IfcPropertySets' to accommodate project-specific data. This approach allows construction stakeholders to ensure that valuable project information is not overlooked or lost and can be utilized for progress tracking and analysis.

To enable the communication of additional progress information, such as textual comments and estimated finish dates, IfcPropertySets can be created using Algorithm 3. The name, description, and value of the IfcPropertySets can be customized for the specific types of information being stored. For example, an IfcPropertySet can be created for progress comments with the name 'Progress Comments' and a value in the IfcText type. Similarly, an IfcPropertySet can be created for estimated finish dates with the name 'EstimatedFinish' and a nominal value (e.g., 2022-12-27T14:57:48.803492) set as a normal date in the IfcDateTime representation.

Algorithm 3: Creating IFC-based entities for additional progress information into IFC-based BIM

Input: IFC-based BIM, Additional information details (Name, Description and Value)

Output: Updated IFC-based BIM

```

1  for each 'IfcBuiltElementi' in IFC-based BIM do
2      create an 'IfcPropertySingleValue' entity
3      Input Name, Description and Value into 'IfcPropertySingleValue' attributes Name, Description, Nominal Value
4      create an 'IfcPropertySet' entity
5      link 'IfcPropertySet' with 'IfcPropertySingleValue' as HasProperties
6      create relationship 'IfcRelDefinesByProperties' to assign 'IfcPropertySet' with 'IfcBuiltElement'
7  end for

```

The algorithm creates an 'IfcPropertySingleValue' entity with attributes such as name, description, and NominalValue to store additional information associated with it and later

assign it to the building components. This allows for any information that is not represented in the native IFC schema to be stored in the *IfcPropertySets* of the required components. By utilizing *IfcPropertySets* and *IfcPropertySingleValues*, all relevant progress-related information can be accurately documented and made available for analysis in BIM models.

3.2. Inputting the Planned Progress Information into IFC-Based BIM

Once the progress-related entities are created and linked in the IFC-based BIM model, the proposed method inputs the planned information into the task and the building component-based entities of the BIM model. The planning information mainly includes the time and cost schedules of planned tasks. To achieve this, the planned parameters of tasks are stored in the relevant entities, followed by the estimation and updating of relevant building components. This process enables the updating of building component parameters, which are essential for tracking project progress. The current stage is designed to only enrich those planned entities that do not require regular updating. These entities, which are also shown in Figures 6 and 8, are detailed in Table 1.

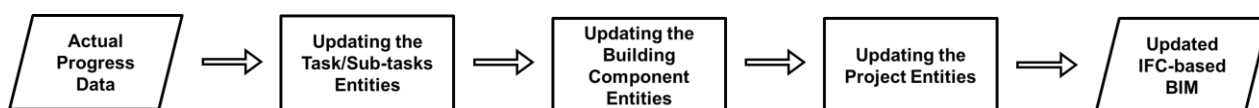


Figure 8. An example of IFC content.

Table 1. Details of planned progress entities.

Level of Information	IFC Entity	Attribute/Property	Data Type	Description
Task(s)/SubTask(s)	IfcTaskTime	ScheduleStart	IfcDateTime	Schedule start time of the task/subtask
		ScheduleFinish	IfcDateTime	Schedule finish time of the task/subtask
		ScheduleDuration	IfcDuration	Schedule duration of the task/subtask
		StatusTime	IfcDateTime	Time at which the status of the task/subtask was last updated
	IfcConstructionMaterialResource	BaseCosts	IfcMonetaryMeasure	Planned cost of the task
Building Component	IfcTaskTime	ScheduleStart	IfcDateTime	Schedule start time of the building component
		ScheduleFinish	IfcDateTime	Schedule finish time of the building component
		ScheduleDuration	IfcDuration	Schedule duration of the building component
		StatusTime	IfcDateTime	Time at which the status of the building component was last updated
	IfcCostItem	CostValues	IfcMonetaryMeasure	Planned cost of building component

To perform this, the required planning information from the construction site is acquired in the form of a list that includes the respective schedule information. The time-related planned information includes the start and finish dates of tasks along with their duration, while the cost-related information lists the total cost planned for the completion of the tasks.

This information is then inputted into BIM using the appropriate IFC type through a program that identifies the tasks from the list using the name and then enriches the corresponding entities or attributes of the corresponding task present in the BIM with the planned information. From the corresponding task in the schedule list, the program later infers all the 'IfcTaskTime' and 'IfcConstructionMaterialResource' that are the sub-types of the 'IfcTask' entity and inputs their attributes.

The algorithm for the enrichment of planned entities is explained in Algorithm 4. Initially, the task-level entities are inputted from the given planned information, i.e., time and cost-related parameters of the task from the schedule list. The start date, finish date, and duration of each task are enriched into the relevant attributes (ScheduleStart, ScheduleFinish, ScheduleSDuration) of the 'IfcTaskTime' entity under the corresponding 'IfcTask'. Similarly, the task cost is stored in the attribute (AppliedValues) of the 'BaseCosts' entity under 'IfcConstructionMaterialResource', which is linked with the corresponding 'IfcTask'. The information regarding the task is critical for the timely completion of the building and is directly stored in the corresponding 'IfcTask' entity. In the case of building components containing several subtasks, the main task duration is also updated based on the attributes of the sub-task by using the following Equations (1)–(3):

Algorithm 4: Inputting planned information into IFC-based BIM

Input: IFC-based BIM, Planned time-related Schedule, Planned cost Schedule
Output: Updated IFC-based BIM

```

1  for each 'IfcBuiltElementi' in IFC-based BIM do
2    if sub-activities exist then
3      for each sub-activity do
4        Input the attributes of 'IfcTaskTimen' entity under 'IfcTaskn' entity
5        Input the attributes of 'IfcCostValuen' entity under 'IfcConstructionMaterialResourcenPlanned' entity
6      end for
7      Update the attributes of 'IfcTaskTimei' entity under 'IfcTaski' entity
8    else
9      Input the attributes of 'IfcTaskTimei' entity under 'IfcTaski' entity
10     Input the attributes of 'IfcCostValuei' entity under 'IfcConstructionMaterialResourceiPlanned' entity
11   end if
12   Update the attributes of 'IfcCostValuei' entity under 'IfcCostItemiPlanned' entity
13 end for
  
```

$$IfcTaskTime_i \cdot ScheduleStart = \min[IfcTaskTime_n \cdot ScheduleStart]_{n=1,2,3,\dots,N} \quad (1)$$

$$IfcTaskTime_i \cdot Schedulefinish = \max[IfcTaskTime_n \cdot ScheduleStart]_{n=1,2,3,\dots,N} \quad (2)$$

$$IfcTaskTime_i \cdot ScheduleDuration = IfcTaskTime_i \cdot Schedulefinish - IfcTaskTime_i \cdot ScheduleStart \quad (3)$$

In the above equations, $IfcTaskTime_n$ and $IfcTaskTime_i$ represent the 'IfcTaskTime' entity of sub-task 'n' and main task 'i' associated with the building component, respectively, while min and max are the minimum and maximum parameters from the N number of total sub-tasks having attributes: *ScheduleStart*, *ScheduleFinish*, and *ScheduleDuration*.

In the end, the 'IfcCostItem' entity representing the cost of the building component is also estimated for updating using Equation (4). It is based on the total costs of all the task(s) that are required for the completion of a building component, and the task costs are stored under the 'IfcConstructionMaterialResource' entity.

$$IfcCostItem_i^{Planned} \cdot CostValues = \begin{cases} \sum_{n=1}^N (IfcConstructionMaterialResource_n^{Planned} \cdot BaseCosts) & \text{if subactivity exist} \\ IfcConstructionMaterialResource_i^{Planned} \cdot BaseCosts & \text{else} \end{cases} \quad (4)$$

In the above equation, $IfcConstructionMaterialResource_i^{Planned}$ and $IfcConstructionMaterialResource_n^{Planned}$ are the planned resource entities of *i*-th task and the *n*-sub task respectively whereas $IfcCostItem_i^{Planned}$ is the entity representing the total planned cost of the *i*-th building component.

The values of the progress parameters are stored in their relevant entities in the form of suitable data types. For example, 'ScheduleStart' and 'ScheduleFinish' attributes should be updated with values having the IfcDateTime data type, which has the representation of YYYY-MM-DDThh:mm:ss defined by ISO 8601. The data types of other planned entities are given in the column 'Data type' of Table 1.

These entities are those that do not update regularly. Hence, the current stage is only performed if these entities are empty, which is possible for the first time. This stage is

designed to only input the planned entities and does not need to be executed every time if the planned entities are already there.

3.3. Updating the Actual Progress into IFC-Based BIM

The objective of this stage is to enrich the actual progress-related IFC entities in the BIM model based on the progress information obtained from the construction site. It not only updates the task and building component-based entities according to the given actual progress but also revises the project-based entities to express the overall building progress, as shown in Figure 8. The updating stage mainly involves inferring the required entities, estimating their values, and then adding progress values to them.

The actual information includes the progress of completion (or completion ratios) and the actual cost of the tasks that are either under progress or completed, along with the inspection date. This actual information is sufficient to estimate the concerned attributes and the respective entities that are being updated. The details of the actual progress-related entities, also shown in Figures 6 and 8, are given in Table 2.

Table 2. Details of actual progress entities.

Level of Information	IFC Entity	Attribute/Property	Data Type	Description
Task(s)/SubTask(s)	IfcTaskTime	Completion	IfcPositiveRatioMeasure	Completion ratio of the task/subtask
		ActualStart	IfcDateTime	Actual start time of the task/subtask
		ActualFinish	IfcDateTime	Actual finish time of the task/subtask
		ActualDuration	IfcDuration	Actual duration of the task/subtask
		StatusTime	IfcDateTime	Time at which the status of the task/subtask was last updated
	IfcConstructionMaterialResource*	BaseCosts	IfcMonetaryMeasure	Planned cost of the construction material resource
	IfcConstructionMaterialResource*	BaseCosts	IfcMonetaryMeasure	Base cost of the construction material resource
Building Component	IfcTaskTime	Completion	IfcPositiveRatioMeasure	Completion ratio of the building component
		ActualStart	IfcDateTime	Actual start time of the building component
		ActualFinish	IfcDateTime	Actual finish time of the building component
		ActualDuration	IfcDuration	Actual duration of the building component
		StatusTime	IfcDateTime	Time at which the status of the building component was last updated
	IfcTask	IsMilestone	IfcBoolean	Indicates if the task is a milestone
		Status	IfcLabel	Status of the task (NOTSTARTED/ STARTED/ COMPLETED)
	IfcPropertySet	EstimatedFinish	IfcDateTime	Estimated finish time of the property set
	IfcCostItem*	CostValues	IfcMonetaryMeasure	Cost value of the cost item
Overall Project	Pset_ConstructionResource	ScheduleCompletion	IfcNormalisedRatioMeasure	Planned completion ratio of the overall project
		ActualCompletion	IfcNormalisedRatioMeasure	Actual completion ratio of the overall project
		ScheduleCost	IfcMonetaryMeasure	To-date scheduled cost of the overall project
		ActualCost	IfcMonetaryMeasure	To-date actual cost of the overall project

The procedure to update the IFC-based BIM model in this stage is outlined in Algorithm 5, while the estimation required for this updating, along with the procedure for all three steps, is detailed below:

Algorithm 5: Updating actual information into IFC-based BIM

Input: IFC-based BIM, Actual progress Information
Output: Updated IFC-based BIM

```

1  for each 'IfcBuiltElementi' in IFC-based BIM do
2    if sub-activities exist then
3      for each sub-activity do
4        Update the attributes of 'IfcTaskTimen' entity under 'IfcTaskn' entity
5        Update the attributes of 'IfcCostValuen' entity under 'IfcConstructionMaterialResourcenActual' entity
6      end for
7    else
8      Update the attributes of 'IfcTaskTimei' entity under 'IfcTaski' entity
9      Update the attributes of 'IfcCostValuei' entity under 'IfcConstructionMaterialResourceiActual' entity
10     Update the attributes of 'IfcCostItemiPlanned' entity
11     Update the attributes of 'IfcCostItemiPlanned' entity
12     Update the attributes of 'IfcTaski' entity
13     Update the propertyset 'EstimatedFinishDatei' entity
14   end for
15   Update the propertyset 'Pset_ConstructionResource' entity

```

3.3.1. Tasks/Sub-Task

As the planned information for tasks/subtasks is already inputted into the IFC-based BIM, the actual progress entities are estimated based on actual progress to update fully completed or partially completed tasks. For time-related updates, the 'IfcTaskTime' entity is updated with different attributes related to actual progress, such as 'Completion', 'Actual-Start', and 'ActualFinish', while the resource entity 'IfcConstructionMaterialResource' with attribute ObjectType = 'Actual' is designated for the actual cost. Algorithm 6, as shown below, estimates the actual attributes based on the completion ratio of each task using their 'IfcTaskTime' entity.

Algorithm 6: Updating IfcTaskTime entity using actual progress information

Input: 'IfcTaskTime' entity, Time-related progress information
Output: Updated IFC-based BIM

```

1  If IfcTask.Tasktime.ActualStart = Nil then
2    if IfcTask.Tasktime.Completion > 0 and IfcTask.Tasktime.Completion < 1 then
3      IfcTask.Tasktime.ActualStart = InspectionDate
4      IfcTask.Tasktime.Statustime = InspectionDate
5    end if
6  end if
7  If IfcTask.Tasktime.ActualFinish = Nil then
8    if IfcTask.Tasktime.Completion = 1 then
9      IfcTask.Tasktime.ActualFinish = InspectionDate
10     IfcTask.Tasktime.ActualDuration = IfcTask.Tasktime.ActualFinish - IfcTask.Tasktime.ActualStart
11     IfcTask.Tasktime.Statustime = InspectionDate
12   end if
13 end if

```

3.3.2. Building Components

After updating the entities of individual tasks, the progress entities of individual building components are processed. This includes updating the 'IfcTaskTime' and 'IfcCostItem' entities associated with each component according to the estimated parameters from tasks. If that building component contains only one task, then this task will act as the main task, having attributes already updated in the last step. However, if there are subtasks in the building component, then the actual attributes of the 'IfcTaskTime' entity will be estimated. The actual start, finish, and duration attributes are estimated similarly to the planned attributes in Equations (1)–(3). Later, the completion of the building component is estimated using the following Equation (5), which takes the weighted average of the included tasks based on their schedule duration:

$$Completion_i^{Actual} = \frac{\sum (IfcTaskTime_n \cdot Completion \times IfcTaskTime_n \cdot ScheduleDuration)}{\sum (IfcTaskTime_n \cdot ScheduleDuration)} \quad (5)$$

In the above equation, $Completion_i^{Actual}$ is the actual completion ratio of particular building component 'i' whereas $Completion$ and $ScheduleDuration$ are representing the completion ratio and schedule duration of sub-task 'n', respectively. Based on the estimated completion value, the 'Status' attribute of 'IfcTask' associated with the building component

is also updated. It is set as 'NOTSTARTED' and 'COMPLETED' for completion values of 0 and 1, respectively, while "STARTED" is for values in between. In the end, the critical path method is performed using all the time-related information about building components to calculate the building components that are critical for completion. Correspondingly, these building components were marked as critical by updating the "IsMilestone" of their associated 'IfcTask' entity to "True" and the non-critical components are marked as "False".

Regarding the cost of a building component, already created entities 'CostItem' for actual progress under each 'IfcBuiltElement' are updated based on the cost values of individual tasks represented with the 'IfcConstructionMaterialResource'. The actual cost is estimated using the following Equation (6):

$$IfcCostItem_i^{Actual}.CostValues = \sum_{n=1}^N (IfcConstructionMaterialResource_n^{Actual}.BaseCosts \times Completion_n) \quad (6)$$

In the above equation, $IfcCostItem_i^{Actual}$ Represents the actual cost of building component 'i' that has been spent to date, whereas $IfcConstructionMaterialResource_n^{Actual}$ represents the actual cost of sub-task 'n' in that component with a completion ratio $Completion_n$. The attributes CostValues and BaseCost of both entities are linked to 'Ifc-CostValue' to represent their cost.

In the end, the estimated finish date is also calculated for each building component 'i' using time attributes based on the actual progress rate of all the individual tasks involved in that component, using the following Equation (7):

$$EstimatedFinishDate_i = IfcTaskTime_i \cdot ActualStart + \frac{(InspectionDate - IfcTaskTime_i \cdot ActualStart)}{IfcTaskTime_i \cdot Completion} \quad (7)$$

3.3.3. Project

After updating the entities of all the individual building components with progress information, the overall progress of the project is estimated for updating the supported IFC entity. It includes the up-to-date total completion and cost parameters for both planned and actual progress. These project-based parameters are updated in the native IfcProperty-Set 'Pset_ConstructionResource' defined for storing the parameters to track the project's progress over time. In this property, the to-date planned and actual completion progress is stored in properties 'ScheduleCompletion' and 'ActualCompletion', whereas the to-date total planned and actual cost are stored under 'ScheduleCost' and 'ActualCost', respectively.

Initially, the parameters for up-to-date planned and actual completion are estimated based on the individual progress parameters of building components. To date, actual completion of the project can be calculated using Equation (8), which considers the weighted average of individual completion ratios of building components for actual progress. Similarly, the to-date planned completion ratio of the project is calculated using Equation (9), which takes into account the overall time-related project parameters.

$$To - dateActualCompletion = \frac{\sum (IfcTaskTime_i \cdot Completion \times IfcTaskTime_i \cdot ActualDuration)}{\sum (IfcTaskTime_i \cdot ActualDuration)} \quad (8)$$

$$To - datePlannedCompletion = \frac{(InspectionDate - \min[IfcTaskTime_i \cdot ScheduleStart]_{i=1,2,3,...,I})}{\max[IfcTaskTime_i \cdot ScheduleFinish]_{i=1,2,3,...,I} - \min[IfcTaskTime_i \cdot ScheduleStart]_{i=1,2,3,...,I}} \quad (9)$$

In the above equations, $IfcTaskTime_i$ is the time-related progress entity of building component 'i' having attributes *ScheduleStart*, *ScheduleStart*, *ActualDuration* whereas *InspectionDate* represents the date of progress inspection performed at the construction site.

After that, the to-date total planned and actual costs are estimated. The cost item entities are defined in the project $IfcCostItem_t^{Planned}$ and $IfcCostItem_t^{Actual}$ represent the sums of cost items of individual building components due to the nested relationship. As all the planned costs are already inputted while the actual costs are only added with time

as per the actual progress, hence, $IfcCostItem_t^{Planned}$ and $IfcCostItem_t^{Actual}$ entities are representing the total planned cost and to-date actual cost (total), respectively. Therefore, the individually planned completion of building components until the inspection date is initially estimated using Equation (10), and then, based on that to-date total planned cost of the project, it is computed using Equation (11).

$$Completion_i^{Planned} = \begin{cases} 1 & \text{if } InspectionDate > IfcTaskTime_i \cdot ScheduleFinish \\ \frac{(InspectionDate - IfcTaskTime_i \cdot ScheduleStart)}{IfcTaskTime_i \cdot ScheduleDuration} & \text{elif } InspectionDate > IfcTaskTime_i \cdot ScheduleStart \\ 0 & \text{else} \end{cases} \quad (10)$$

$$To - date \text{ planned Cost} = \Sigma \left(IfcCostItem_i^{Planned} \cdot CostValue \times Completion_i^{Planned} \right) \quad (11)$$

$$To - date \text{ ActualCost} = IfcCostItem_t^{Actual} \cdot CostValue \quad (12)$$

In the above equation, ScheduleStart, ScheduleFinish, and ScheduleDuration are the attributes of the 'IfcTaskTime' entity associated with the main task 'IfcTask' of building component 'i' to estimate its completion ratio $Completion_i^{Planned}$ till inspection date, which is represented as *InspectionDate*. Similarly $IfcCostItem_t^{Planned}$ and $IfcCostItem_t^{Actual}$ are representing the cost item nest with the individual cost items of building component 'i'. It is pertinent to mention here that the proposed method does not create new entities for time- or cost-related information for the overall project as the project IfcPropertyset (Pset_ConstructionResource) already has the entities of these project-based progress information.

After estimating all these parameters, they are updated into their respective entities with appropriate data types as given in Table 2. The use of proper and relevant data types and units of measure for each attribute is crucial in ensuring the accuracy of the information added to the IFC-based BIM model.

3.4. Reporting the Progress from Updated IFC-Based BIM

The proposed method not only updates progress information in the BIM model but also extracts this information for efficient and user-friendly progress reporting. The extracted information can be used to assess progress by comparing the planned progress information with actual information. Additionally, the extracted progress parameters are processed based on their combination to find the earned value parameters for valuable insights about progress to ensure the project stays within budget and on schedule. Moreover, this stage makes use of BIM information to create more automated representations of extracted progress through 2D charts and 3D models, providing a clearer visualization of the project than traditional reporting. Overall, this approach enhances project monitoring by utilizing BIM technology to provide more accurate and detailed progress tracking.

The updated progress information is retrieved by accessing the corresponding IFC entities that were processed during the update. These entities are organized in the standard hierarchical structure of the IFC schema, which provides a clear path to access them. The proposed method links the progress-related entities to their corresponding building components. For example, for a particular building component 'IfcSlab', the time-related progress information is accessed through the subtype 'IfcTaskTime' entity of 'IfcTask' with linkage (IfcTask → IfcRelAssignsToProduct → IfcSlab), while the cost information is accessed through the subtype entity 'IfcCostValue' of 'IfcCostItem' respectively with linkage (IfcCostItem → IfcRelAssignsToControl → IfcSlab). The building component entities (e.g., IfcSlab, IfcWall, IfcDoor, etc.) are the subtypes of 'IfcBuildingElement' that are present under the hierarchy (IfcBuiltElement → IfcElement → IfcProduct → IfcObject → IfcObjectDefinition → IfcRoot) in the IFC schema. The extracted progress information from the entities is later used to report the progress monitoring of the construction project. In this way, information is directly retrieved from the concerned entities in the same way they were accessed to store information, which ultimately ensures the accurate and reliable communication of information within BIM.

The retrieved progress information can be further utilized to perform the earned value analysis, which is a well-accepted and common progress measurement in the construction industry [76]. The earned value analysis provides additional progress indicators that act as an early warning of performance problems. Integrating this analysis into the current method can facilitate the reporting stage. The updated BIM model contains both cost and schedule information, which is stored in the appropriate work breakdown structure in the BIM model being linked through the IFC schema. This presence of information entities in this standardized form enables the computation of earned value indicators such as variances (cost/schedule, variance at completion), performance indices (cost/schedule performance index), and forecasts (estimate at completion, estimate to completion). The proposed method also performs the earned value analysis of individual building components to analyze each component for comprehensive project tracking. The approach to performing the earned value analysis is illustrated in Figure 9.

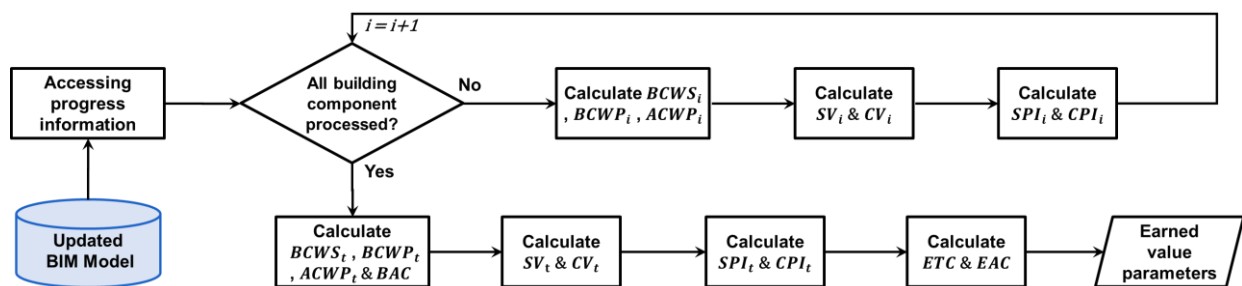


Figure 9. Flowchart to estimate the earned value parameters.

Initially, the earned value measures are calculated, which include the budgeted cost of work scheduled (BCWS), the budgeted cost of work performed (BCWP), and the actual cost of work performed (ACWP). The formulas to calculate these measures for i -th building components using their IFC entities are given in Equations (13)–(15).

$$BCWS_i = IfcCostItem_i^{Planned} \cdot CostValues \times Completion_i^{Planned} \quad (13)$$

$$BCWP_i = IfcCostItem_i^{Planned} \cdot CostValues \times Completion_i^{Actual} \quad (14)$$

$$ACWP_i = IfcCostItem_i^{Actual} \cdot CostValue \quad (15)$$

After calculating these measures, the variances and the performance indices are calculated. The variances, including the schedule variance ($SV_i = BCWP_i - BCWS_i$) and cost variance ($CV = BCWP_i - ACWP_i$) are calculated. Similarly, the performance indices including the schedule performance index ($SPI_i = BCWP_i / BCWS_i$) and cost performance index ($CPI_i = BCWP_i / ACWP_i$) are calculated.

After calculating the earned value parameters for the individual building components, earned value analysis at the project level is performed, which includes the calculation of earned value parameters based on the progress parameters of the overall project. The earned value measures for the project are computed by summing up all the measures of building components, including $BCWS_t = \sum_{i=1}^I (BCWS_i)$, $BCWP_t = \sum_{i=1}^I (BCWP_i)$, $ACWS_t = \sum_{i=1}^I (ACWP_i)$. Additionally, the budget at completion ($BAC = \text{Total Planned Cost}$) is also calculated. After that, the variances and performance indices are calculated using these cumulative measures. Furthermore, the forecast parameters are also calculated, including the 'Estimate To Complete' ($ETC = EAC - ACWP_t$) and 'EstimateAtCompletion' ($EAC = ACWP_t + ((BAC - BCWP_t) / CPI_i)$) values describing the estimated cost to complete remaining works and the estimated total cost required for completion, respectively. These calculated earned value parameters can also be updated into an IFC-based BIM using `IfcPropertySet` if required.

The progress parameters extracted from the updated BIM model, along with the estimated additional parameters, can be utilized to report the construction progress of the project. However, additional efforts were made to further exploit the capabilities of the BIM model to facilitate the reporting process and make it more user-friendly. By leveraging the BIM model's rich (non-progress-related) information, progress information can be reported more efficiently and accurately using various techniques such as bar charts, Gantt charts, and 3D model visualization. This approach not only enhances the visual presentation of the progress but also provides a better understanding of the project's status to all the stakeholders involved. Table 3 provides details on the different types of reporting utilized for different types of progress information to report construction progress. It is worth noting that this is solely based on the BIM information without relying on any third-party software.

Table 3. Details of different reporting types for different categories of progress information.

Category of Progress Information	Type of Reporting	Purpose
Time-related progress information	3D color-coded visualization	Provides a quick overview of time performance and highlights areas where the project is ahead/behind schedule for individual building components and the overall project.
	3D visualization of model	Allows for visual inspection of progress and identification of potential issues for individual building components and the overall project
	2D Gantt chart	Shows the overall project schedule, critical path, and delays for individual building components and the overall project
Cost-related progress information	3D color-coded visualization	Provides a quick overview of cost performance and highlights areas where costs are over/under budget for individual building components and the overall project
	2D bar chart	Allows for a detailed comparison of planned versus actual costs for individual building components and overall project
Earned value parameters	3D color-coded visualization	Provides a quick overview of earned value indicators and highlights components where indicators are over/under budget for individual building components
	Table form	Provides a detailed analysis of project performance by comparing planned versus actual costs and schedule for individual building components and overall project
	Line graph	Shows the trend of earned value parameters over time for individual building components and overall project
Additional information-estimated finish date	Textual form	Provides a simple and easy-to-understand summary of project status and estimated completion date for individual building components and overall project
	3D visualization of model	Shows the expected completion of activities in the coming week/month for individual building components
Additional information-textual comments	Textual form	Allows for a detailed explanation of project status and any issues that may be affecting progress for individual building components and overall project

An automated program was developed to extract the necessary information on building components and generate the appropriate construction progress monitoring reports. Time-related progress information is extracted from the entities *IfcTaskTime_i*, *Pset_ConstructionResource*, and the cost-related information is extracted from entities *IfcCostItem_i^{Planned}*, *IfcCostItem_i^{Actual}*, *Pset_ConstructionResource*. Earned value parameters and additional information are extracted from *IfcPropertySet*. To create the 2D charts, graphs, and textual information, the relevant

progress parameters are extracted into a suitable arrangement accordingly. To perform the 3D visualization, the geometrical information of each building component is extracted from the BIM model using the different subtypes of geometrical entities such as 'IfcRepresentation', 'IfcGeometricRepresentationItem', 'IfcSpatialElement', and 'IfcOpeningElement'. Subsequently, a 3D mesh model is created based on the geometrical information and as per the required objective, such as showing only the 3D mesh of the planned or actual completed building components or color-coding components according to their progress indicators obtained from comparison or earned value parameters.

In the end, a web-based application was developed to allow for real-time reporting of all this progress information from the BIM model in the recommended format of texts, 2D charts, and 3D models. The application enables users to view the progress information of building components at any given time and to track the project's progress status efficiently. This approach enhances the visual presentation of progress information and provides stakeholders with a better understanding of the project's status.

4. Results and Discussion

The proposed method was tested on several IFC-based BIM models to assess the exchange of progress information. A program was developed in Python based on the proposed methodology for testing. A BIM model, as shown in Figure 10, is a standard 3D BIM without any additional IFC entities, making it ideal for testing, which focuses on incorporating progress-related entities. This building model is also used in other studies [3,5] in detail.

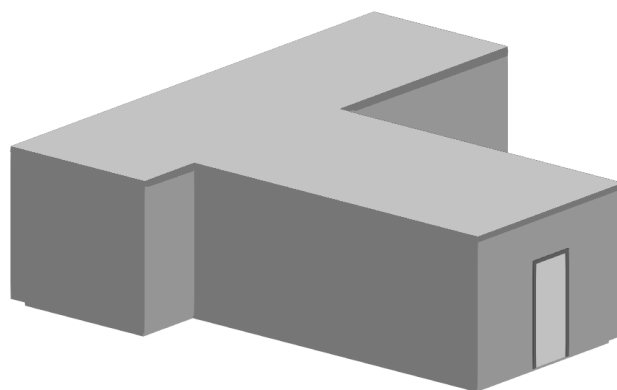


Figure 10. Geometrical visualization of an IFC-based BIM model illustrating various building components.

To perform the BIM updating effectively, the actual progress information was obtained after the Scan-vs-BIM comparison. The proposed method was applied to the original IFC-based BIM model to obtain the updated model integrated with the progress entities according to the planned progress information, including time, cost, and other additional parameters. A segment of the IFC code snippet is shown in Figure 11 from the updated BIM model, in which some newly integrated progress entities of the building component (#197) are populated with the planned and actual progress parameters. If we examine the given code, then it is evident that the new entities were created according to the proposed framework. The time-related parameters stored in the 'IfcTaskTime' entity (#2269) being linked to its corresponding task entity (#2270) are present along with the other time-related entities. Similarly, the planned and actual cost entities (IfcCostItem), represented by #2339 and #2394, respectively, linked to their respective cost schedules (#2265, #2267) are brought to the forefront. These time and cost-related entities, through their respective task (#2270) and schedule (#2265, #2267) entities, are linked to their corresponding building component (#197) through relationship entities (#2271, #2266, #2268).

[illegible]

Figure 11. IFC code snippet expressing the newly integrated IFC entities.

Figure 12 shows the progress information related to that building component (#197), retrieved from the BIM model, which is reported in the Python environment. The estimated actual progress parameters, related to time, cost, and other additional information in the form of an expected finish date, corresponding to the approximately 56% completion of the component, are also presented. Although the actual costs were relatively high, the estimated actual cost for the component (#197) is lower, mainly due to the fact that the component is more than half built. Furthermore, the earned value indicators are also estimated to individually analyze each component for improved project tracking. The performance indices of the component (#197) are less than 1 for both schedule and cost, highlighting how much the component is behind/over the planned schedule/cost.

```

#197=IfcWall('3zv0SswuRX5rhB2K73V3BRr',#42,'Wall 1',$,'Basic Wall:Generic - 200mm',#158,#191,'354100',.NOTDEFINED.)
..... Extracting details
GlobalId:                3zv0SswuRX5rhB2K73V3BRr
Building Component Name:  Wall 1
Schedule Start Date :    2022-09-12T00:00:00
Schedule Finish Date :   2022-09-16T00:00:00
Planned Cost :           264.00000000000364
Status Time :            2022-09-22T00:00:00
Completion Ratio :       0.5626106364965751
Actual Start Date :      2022-09-19T00:00:00
Actual Finish Date :     None
Expected Finish Date :   2022-09-25T00:00:00
Actual Cost              204.22766104825953
Schedule Performance Index (SPI):0.5626106364965751
Cost Performance Index (CPI):   0.7272727272727274

```

Figure 12. Retrieval of progress information for the building component (#197) using Python.

Furthermore, the same updated BIM file was exported into the external IFC-based software to confirm the successful storage and accessibility of information with the other applications. The building component (#197) along with its attributes are highlighted in Figure 13, where the BIM model was exported in 'Open IFC Viewer version 23.3.0 [77], as also used in another study [78]. The parameter 'EstimatedFinish' is also noticeable, which was stored under the property set mainly because this additional information did not fit into the standard entities of the IFC schema. The successful export of the updated BIM model in the external software also confirms that the proposed method performed the successful update according to the IFC schema, as minor deviation while creating and updating IFC entities can corrupt the complete IFC file.

In the end, the progress results were also color-coded according to the completion ratios of building components to assess the progress made. For this purpose, the geometrical information of the components is extracted into mesh form, and then four colors are assigned based on their completion ratios: 0–30% (red), 30–50% (yellow), 50–70% (orange), 70–90% (green), and 90–100% (blue). The color-coded models of the updated BIM model are illustrated in Figure 14 from different viewpoints.

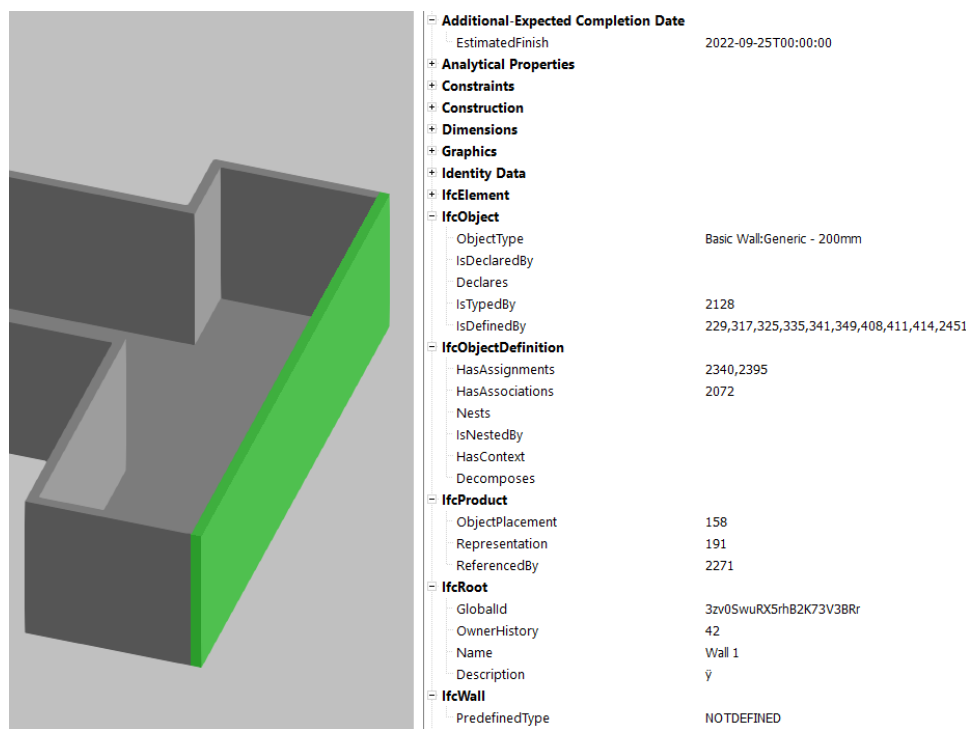


Figure 13. Building component (#197) highlighted in the exported BIM model, along with its attribute details, viewed in the external software (Open IFC Viewer, version 23.3.0).

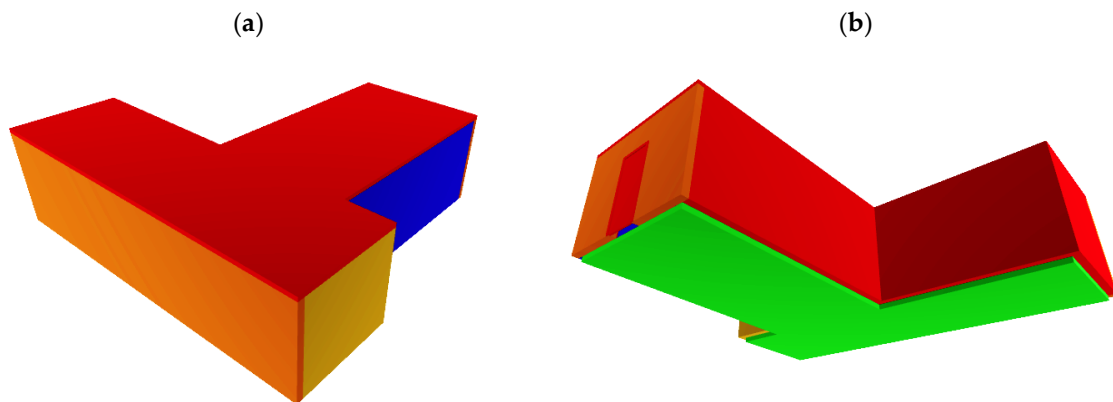


Figure 14. Visualization of building components using different colors (red, yellow, orange, green, and blue) corresponding to their respective completion ratios (0–30%, 30–50%, 50–70%, 70–90%, and 90–100%) in two different viewpoints (a,b).

Furthermore, a web-based application was also developed to effectively report the progress information from the updated IFC-based BIM model. The application only requires the updated BIM model and then details all the progress-related information in real-time through the different types of reporting. A screenshot of the application is shown in Figure 15, displaying the project-level progress details of the updated BIM model in the form of charts, graphs, and 3D visualization. The developed application makes use of different progress parameters present within the standardized hierarchy of IFC-based BIM to facilitate reporting. It is pertinent to mention here that the proposed method does not store the individually earned parameter values, mainly to ensure a possible reduction of the IFC file size. The parameters are computed in real-time based on the already-stored schedule and cost parameters; however, they can be stored in a separate property set if required.

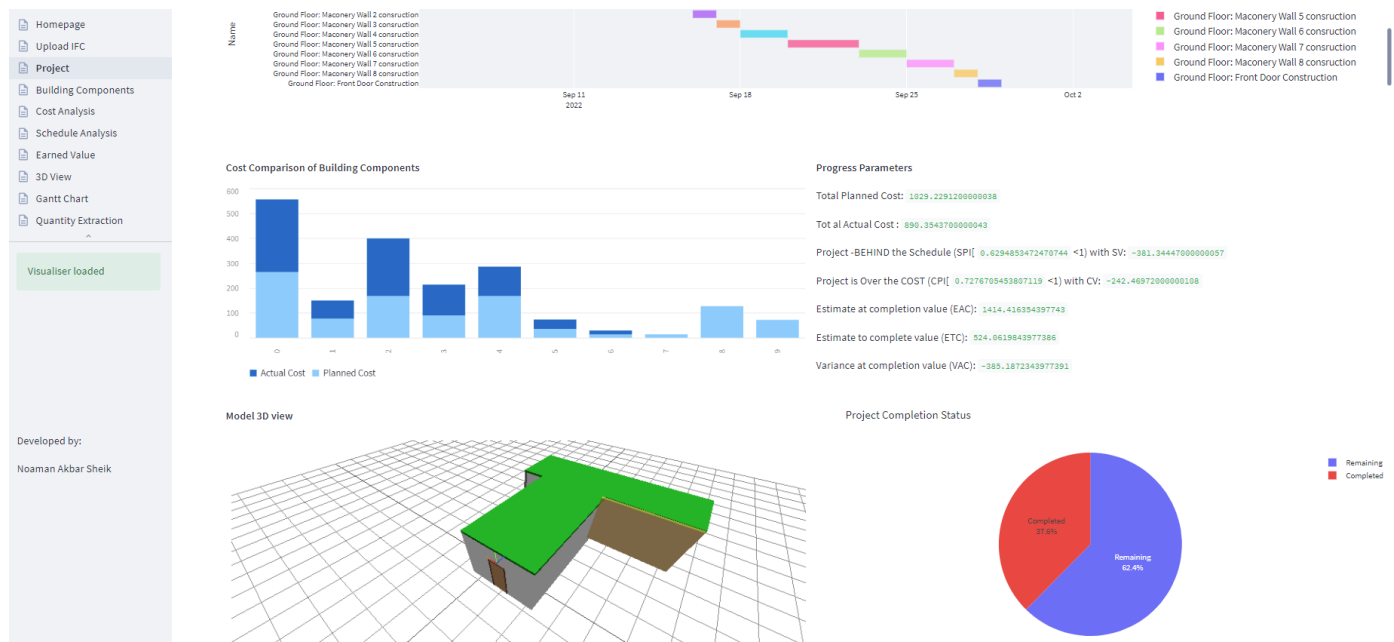


Figure 15. A screenshot of the developed web-based application demonstrating the project-level progress details of the updated BIM model.

The effectiveness of the proposed method was further evaluated by applying it to other IFC-based models, including the ISPRS model studies dataset [79]. In the evaluation, random progress information was exchanged with the BIM models to access its updating framework, as the proposed method enables the incorporation of input data from various sources. The method successfully updated the progress information for these models as well. Figure 16 shows the visualization of the two ISPRS models, in which their building components are color-coded according to their time-related progress details after exchanging progress information.

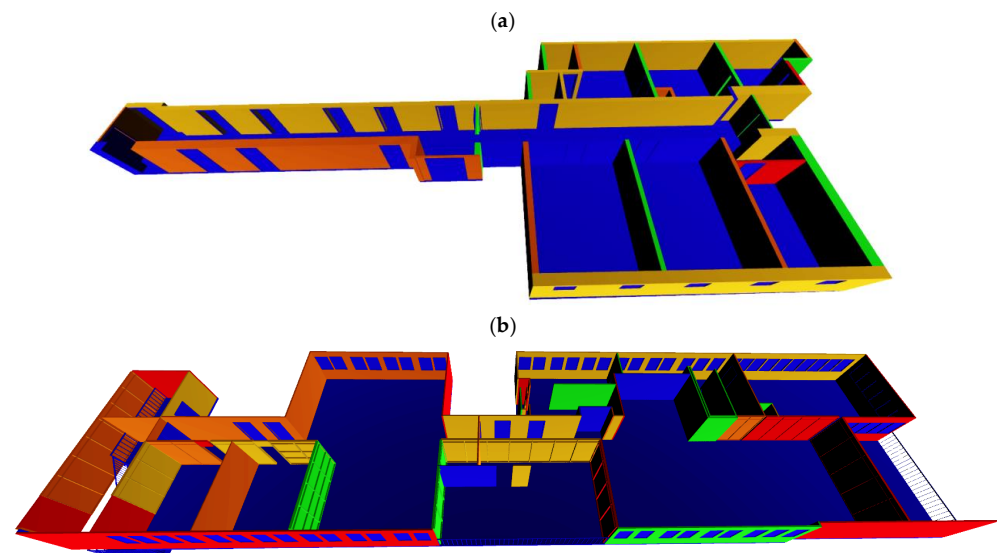


Figure 16. Color-coded visualization of two different BIM models (a,b) highlighting the progress status of their building components: 0–30% (Red), 30–50% (Yellow), 50–70% (Orange), 70–90% (Green), and 90–100% (Blue).

From the results, it is evident that the proposed method is capable of updating IFC-based BIM models with progress information. The method processed the IFC models

by creating the new entities, accurately estimating their progress parameters, updating existing entities, and then successfully reporting the progress information in a user-friendly manner. These IFC-based BIM models, updated with progress information, can be utilized for construction progress monitoring, facilitating the automated transfer of information in construction. Similarly, the development of a web-based application also proved that progress details can be effortlessly reported without any reliance on third-party commercial software if the information is stored in a structurally organized way through a standard framework, eventually leveraging the potential of BIM technology in construction projects. This proposed method holds significant potential for enhancing the efficacy of other construction progress monitoring studies [3,5,42,63] by providing them with a standardized solution for information exchange.

5. Conclusions

BIM provides an extensive and accurate digital representation of a building in the form of geometrical and semantic information that describes its physical and functional characteristics. However, its potential is not fully realized in construction progress monitoring and reporting. IFC is a widely recognized open standard for BIM exchange; however, its utility needs to be addressed for exchanging progress information based on the latest IFC4.x schema, which encompasses not only time-related information but also cost and other non-standardized information. The current study detailed a comprehensive methodology with a task-based approach to exchange the progress information of under-construction buildings with the BIM model using the latest IFC4 schema in four stages. The progress information is integrated into the BIM model through structural organization in the form of appropriate IFC entities. Apart from the standard progress information, including the time-related schedule or cost, the method also allows the integration of additional semantic information that may not fit into the defined IFC entities to ensure that valuable information is not overlooked. To enable effective progress monitoring, the method not only updates the BIM model but also allows the efficient retrieval of progress information, which is reported in the form of additional progress indicators, including earned value analysis, to offer valuable insights. Furthermore, it is also ensured that the estimated progress values of the building components and project are being revised according to their corresponding tasks or sub-tasks.

The testing demonstrated the successful updating of the BIM models to accommodate the progress information while demonstrating a web-based application that reads the intricate information from the IFC entities and then reports it in a user-friendly format. The exchange of progress information using the IFC-based BIM is a significant step towards automation in construction progress monitoring that enables the ability to accurately track construction progress and timely identify deviations, which eventually improves decision-making, increases efficiency, and enhances project management.

Author Contributions: Conceptualization, N.A.S. and G.D.; methodology, N.A.S. and G.D.; validation, N.A.S.; formal analysis, N.A.S.; data curation, N.A.S.; writing—original draft preparation, N.A.S.; writing—review and editing, G.D. and P.V.; supervision, G.D. and P.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hamledari, H.; McCabe, B.; Davari, S. Automated computer vision-based detection of components of under-construction indoor partitions. *Autom. Constr.* **2017**, *74*, 78–94. [[CrossRef](#)]
2. Golparvar-Fard, M.; Pena-Mora, F.; Savarese, S. Automated progress monitoring using unordered daily construction photographs and IFC-based building information models. *J. Comput. Civ. Eng.* **2014**, *29*, 04014025. [[CrossRef](#)]

3. Sheik, N.A.; Veelaert, P.; Deruyter, G. Registration of Building Scan with IFC-Based BIM Using the Corner Points. *Remote Sens.* **2022**, *14*, 5271. [CrossRef]
4. Bosché, F. Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Adv. Eng. Inform.* **2010**, *24*, 107–118. [CrossRef]
5. Sheik, N.A.; Deruyter, G.; Veelaert, P. Plane-Based Robust Registration of a Building Scan with Its BIM. *Remote Sens.* **2022**, *14*, 1799. [CrossRef]
6. Mahami, H.; Nasirzadeh, F.; Hosseinaveh Ahmadabadian, A.; Nahavandi, S. Automated progress controlling and monitoring using daily site images and building information modelling. *Buildings* **2019**, *9*, 70. [CrossRef]
7. Azhar, S.; Carlton, W.A.; Olsen, D.; Ahmad, I. Building information modeling for sustainable design and LEED® rating analysis. *Autom. Constr.* **2011**, *20*, 217–224. [CrossRef]
8. Ghaffarianhoseini, A.; Tookey, J.; Ghaffarianhoseini, A.; Naismith, N.; Azhar, S.; Efimova, O.; Raahemifar, K. Building Information Modelling (BIM) uptake: Clear benefits, understanding its implementation, risks and challenges. *Renew. Sustain. Energy Rev.* **2017**, *75*, 1046–1053. [CrossRef]
9. Doumbouya, L.; Gao, G.; Guan, C. Adoption of the Building Information Modeling (BIM) for construction project effectiveness: The review of BIM benefits. *Am. J. Civ. Eng. Archit.* **2016**, *4*, 74–79.
10. Nawari, N.O. BIM standard in off-site construction. *J. Archit. Eng.* **2012**, *18*, 107–113. [CrossRef]
11. Lee, D.; Lee, S.H.; Masoud, N.; Krishnan, M.; Li, V.C. Integrated digital twin and blockchain framework to support accountable information sharing in construction projects. *Autom. Constr.* **2021**, *127*, 103688. [CrossRef]
12. Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K.; Handbook, B. A guide to building information modeling for owners, managers, designers, engineers and contractors. *BIM Handb.* **2011**, *2*, 147–150.
13. Arayici, Y.; Aouad, G. Building information modelling (BIM) for construction lifecycle management. *Constr. Build. Des. Mater. Tech.* **2010**, *2010*, 99–118.
14. Succar, B. Building information modelling framework: A research and delivery foundation for industry stakeholders. *Autom. Constr.* **2009**, *18*, 357–375. [CrossRef]
15. Azhar, S.; Khalfan, M.; Maqsood, T. Building information modeling (BIM): Now and beyond. *Australas. J. Constr. Econ. Build.* **2012**, *12*, 15–28.
16. ISO 16739-1: 2018; Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries—Part 1: Data Schema. ISO: Geneva, Switzerland, 2018.
17. Poljanšek, M. *Building Information Modelling (BIM) Standardization*; European Commission: Brussels, Belgium, 2017.
18. ISO 29481-1; Building Information Models—Information Delivery Manual—Part 1: Methodology and Format. International Organization for Standardization: Geneva, Switzerland, 2016.
19. Söbke, H.; Peralta, P.; Smarsly, K.; Armbruster, M. An IFC schema extension for BIM-based description of wastewater treatment plants. *Autom. Constr.* **2021**, *129*, 103777. [CrossRef]
20. BuildingSMART International. Available online: <https://www.buildingsmart.org/> (accessed on 1 December 2021).
21. BuildingSMART International. *IFC Overview Summary*; BuildingSMART International: Cambridge, UK, 2018.
22. Yu, K.; Froese, T.; Grobler, F. A development framework for data models for computer-integrated facilities management. *Autom. Constr.* **2000**, *9*, 145–167. [CrossRef]
23. Lam, K.; Wong, N.; Shen, L.; Mahdavi, A.; Leong, E.; Solihin, W.; Au, K.; Kang, Z. Mapping of industry building product model for detailed thermal simulation and analysis. *Adv. Eng. Softw.* **2006**, *37*, 133–145. [CrossRef]
24. Ma, Z.; Teng, M.; Ren, Y. Method of extracting static data for building energy consumption monitoring from BIM. *J. Harbin Inst. Technol.* **2019**, *51*, 187–193.
25. Zhang, J.; Yu, F.; Li, D.; Hu, Z. Development and implementation of an industry foundation classes-based graphic information model for virtual construction. *Comput. Aided Civ. Infrastruct. Eng.* **2014**, *29*, 60–74. [CrossRef]
26. Zhang, J.; Guo, J.; Wang, S.; Xu, Z. Intelligent facilities management system based on IFC standard and building equipment integration. *Tsinghua Univ.* **2008**, *48*, 940–942.
27. Xu, Z.; Wang, X.; Xiao, Y.; Yuan, J. Modeling and performance evaluation of PPP projects utilizing IFC extension and enhanced matter-element method. *Eng. Constr. Archit. Manag.* **2020**, *27*, 1763–1794. [CrossRef]
28. Akinci, B.; Boukamp, F. Representation and integration of as-built information to IFC based product and process models for automated assessment of as-built conditions. In Proceedings of the ISARC 2002: 19th International Symposium on Automation and Robotics in Construction, Gaithersburg, MA, USA, 23–25 September 2002; pp. 543–549.
29. Seo, J.; Kim, I. Industry foundation classes-based approach for managing and using the design model and planning information in the architectural design. *J. Asian Archit. Build. Eng.* **2009**, *8*, 431–438. [CrossRef]
30. Choi, J.; Kim, H.; Kim, I. Open BIM-based quantity take-off system for schematic estimation of building frame in early design stage. *J. Comput. Des. Eng.* **2015**, *2*, 16–25. [CrossRef]
31. Akanbi, T.; Zhang, J.; Lee, Y.-C. Data-driven reverse engineering algorithm development method for developing interoperable quantity takeoff algorithms using IFC-based BIM. *J. Comput. Civ. Eng.* **2020**, *34*, 04020036. [CrossRef]
32. Park, J.; Cai, H.; Dunston, P.S.; Ghasemkhani, H. Database-supported and web-based visualization for daily 4D BIM. *J. Constr. Eng. Manag.* **2017**, *143*, 04017078. [CrossRef]

33. Hamledari, H.; McCabe, B.; Davari, S.; Shahi, A. Automated schedule and progress updating of IFC-based 4D BIMs. *J. Comput. Civ. Eng.* **2017**, *31*, 04017012. [CrossRef]
34. Hamledari, H.; Rezazadeh Azar, E.; McCabe, B. IFC-based development of as-built and as-is BIMs using construction and facility inspection data: Site-to-BIM data transfer automation. *J. Comput. Civ. Eng.* **2017**, *32*, 04017075. [CrossRef]
35. Son, H.; Kim, C.; Kwon Cho, Y. Automated schedule updates using as-built data and a 4D building information model. *J. Manag. Eng.* **2017**, *33*, 04017012. [CrossRef]
36. IFC2x3 to IFC4. Available online: https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/annex/annex-f/ifc2x3-to-ifc4/index.htm (accessed on 1 March 2023).
37. Navon, R. Research in automated measurement of project performance indicators. *Autom. Constr.* **2007**, *16*, 176–188. [CrossRef]
38. Zhang, X.; Bakis, N.; Lukins, T.C.; Ibrahim, Y.M.; Wu, S.; Kagioglou, M.; Aouad, G.; Kaka, A.P.; Trucco, E. Automating progress measurement of construction projects. *Autom. Constr.* **2009**, *18*, 294–301. [CrossRef]
39. Yang, J.; Park, M.-W.; Vela, P.A.; Golparvar-Fard, M. Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future. *Adv. Eng. Inform.* **2015**, *29*, 211–224. [CrossRef]
40. Alizadehsalehi, S.; Yitmen, I. A concept for automated construction progress monitoring: Technologies adoption for benchmarking project performance control. *Arab. J. Sci. Eng.* **2019**, *44*, 4993–5008. [CrossRef]
41. Fang, J.; Li, Y.; Liao, Q.; Ren, Z.; Xie, B. Construction Progress Control and Management Measures Analysis. *Smart Constr. Res.* **2018**, *2*, 1–5. [CrossRef]
42. Kavaliauskas, P.; Fernandez, J.B.; McGuinness, K.; Jurelionis, A. Automation of Construction Progress Monitoring by Integrating 3D Point Cloud Data with an IFC-Based BIM Model. *Buildings* **2022**, *12*, 1754. [CrossRef]
43. Kim, C.; Son, H.; Kim, C. Automated construction progress measurement using a 4D building information model and 3D data. *Autom. Constr.* **2013**, *31*, 75–82. [CrossRef]
44. Golparvar-Fard, M.; Savarese, S.; Peña-Mora, F. Interactive Visual Construction Progress Monitoring with D4 AR—4D Augmented Reality—Models. In Proceedings of the Construction Research Congress 2009: Building a Sustainable Future, Seattle, WA, USA, 5–7 April 2009; pp. 41–50.
45. Omar, H.; Dulaimi, M. Using BIM to automate construction site activities. *Build. Inf. Model. BIM Des. Constr. Oper.* **2015**, *149*, 45.
46. Braun, A.; Tuttas, S.; Borrmann, A.; Stilla, U. A concept for automated construction progress monitoring using bim-based geometric constraints and photogrammetric point clouds. *J. Inf. Technol. Constr.* **2015**, *20*, 68–79.
47. Carrera-Hernández, J.; Levesse, G.; Lacan, P. Is UAV-SfM surveying ready to replace traditional surveying techniques? *Int. J. Remote Sens.* **2020**, *41*, 4820–4837. [CrossRef]
48. Abd-Elmaaboud, A.; El-Tokhey, M.; Ragheb, A.; Mogahed, Y. Comparative assessment of terrestrial laser scanner against traditional surveying methods. *Int. J. Eng. Appl. Sci.* **2019**, *6*, 79–84.
49. Matarneh, S.; Elghaish, F.; Rahimian, F.P.; Dawood, N.; Edwards, D. Automated and interconnected facility management system: An open IFC cloud-based BIM solution. *Autom. Constr.* **2022**, *143*, 104569. [CrossRef]
50. Shirowzhan, S.; Sepasgozar, S.M.; Edwards, D.J.; Li, H.; Wang, C. BIM compatibility and its differentiation with interoperability challenges as an innovation factor. *Autom. Constr.* **2020**, *112*, 103086. [CrossRef]
51. Newman, C.; Edwards, D.; Martek, I.; Lai, J.; Thwala, W.D.; Rillie, I. Industry 4.0 deployment in the construction industry: A bibliometric literature review and UK-based case study. *Smart Sustain. Built Environ.* **2021**, *10*, 557–580. [CrossRef]
52. Petronijević, M.; Višnjevac, N.; Prašćević, N.; Bajat, B. The extension of IFC for supporting 3D cadastre LADM geometry. *ISPRS Int. J. Geo Inf.* **2021**, *10*, 297. [CrossRef]
53. Bortoluzzi, B.; Efremov, I.; Medina, C.; Sobieraj, D.; McArthur, J. Automating the creation of building information models for existing buildings. *Autom. Constr.* **2019**, *105*, 102838. [CrossRef]
54. Mihindu, S.; Arayici, Y. Digital construction through BIM systems will drive the re-engineering of construction business practices. In Proceedings of the 2008 International Conference Visualisation, London, UK, 9–11 July 2008; pp. 29–34.
55. Gerbert, P.; Castagnino, S.; Rothballer, C.; Renz, A.; Filitz, R. *Digital in Engineering and Construction*; The Boston Consulting Group: Boston, MA, USA, 2016; pp. 1–18.
56. Justo, A.; Soilán, M.; Sánchez-Rodríguez, A.; Riveiro, B. Scan-to-BIM for the infrastructure domain: Generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data. *Autom. Constr.* **2021**, *127*, 103703. [CrossRef]
57. Kwon, T.H.; Park, S.I.; Jang, Y.-H.; Lee, S.-H. Design of railway track model with three-dimensional alignment based on extended industry foundation classes. *Appl. Sci.* **2020**, *10*, 3649. [CrossRef]
58. Jones, S.; Laquidara-Carr, D.; Lorenz, A.; Buckley, B.; Barnett, S. The business value of BIM for infrastructure 2017. SmartMarket Report 2017. Available online: <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/finance/us-fas-bim-infrastructure.pdf> (accessed on 1 December 2021).
59. Radke, A.; Wallmark, T.; Tseng, M. An automated approach for identification and resolution of spatial clashes in building design. In Proceedings of the 2009 IEEE International Conference on Industrial Engineering and Engineering Management, Hong Kong, China, 8–11 December 2009; pp. 2084–2088.
60. Borrmann, A.; Berkhahn, V. Principles of Geometric Modeling. In *Building Information Modeling: Technology Foundations and Industry Practice*; Springer: Cham, Switzerland, 2018; pp. 27–41.

61. Borrmann, A.; Beetz, J.; Koch, C.; Liebich, T.; Muhic, S. Industry foundation classes: A standardized data model for the vendor-neutral exchange of digital building models. In *Building Information Modeling: Technology Foundations and Industry Practice*; Springer: Cham, Switzerland, 2018; pp. 81–126.
62. Turkan, Y.; Bosche, F.; Haas, C.T.; Haas, R. Automated progress tracking using 4D schedule and 3D sensing technologies. *Autom. Constr.* **2012**, *22*, 414–421. [\[CrossRef\]](#)
63. Pučko, Z.; Šuman, N.; Rebolj, D. Automated continuous construction progress monitoring using multiple workplace real time 3D scans. *Adv. Eng. Inform.* **2018**, *38*, 27–40. [\[CrossRef\]](#)
64. Chen, L.; Luo, H. A BIM-based construction quality management model and its applications. *Autom. Constr.* **2014**, *46*, 64–73. [\[CrossRef\]](#)
65. de Soto, B.G.; Rosarius, A.; Rieger, J.; Chen, Q.; Adey, B.T. Using a Tabu-search algorithm and 4D models to improve construction project schedules. *Procedia Eng.* **2017**, *196*, 698–705. [\[CrossRef\]](#)
66. Lopez, R.; Chong, H.-Y.; Wang, X.; Graham, J. Technical review: Analysis and appraisal of four-dimensional building information modeling usability in construction and engineering projects. *J. Constr. Eng. Manag.* **2016**, *142*, 06015005. [\[CrossRef\]](#)
67. Li, C.Z.; Xue, F.; Li, X.; Hong, J.; Shen, G.Q. An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction. *Autom. Constr.* **2018**, *89*, 146–161. [\[CrossRef\]](#)
68. Deng, Y.; Gan, V.J.; Das, M.; Cheng, J.C.; Anumba, C. Integrating 4D BIM and GIS for construction supply chain management. *J. Constr. Eng. Manag.* **2019**, *145*, 04019016. [\[CrossRef\]](#)
69. Vieira, R.; Carreira, P.; Domingues, P.; Costa, A.A. Supporting building automation systems in BIM/IFC: Reviewing the existing information gap. *Eng. Constr. Archit. Manag.* **2020**, *27*, 1357–1375. [\[CrossRef\]](#)
70. Yang, B.; Dong, M.; Wang, C.; Liu, B.; Wang, Z.; Zhang, B. IFC-based 4D construction management information model of prefabricated buildings and its application in graph database. *Appl. Sci.* **2021**, *11*, 7270. [\[CrossRef\]](#)
71. Noardo, F.; Arroyo Otori, K.; Krijnen, T.; Stoter, J. An inspection of IFC models from practice. *Appl. Sci.* **2021**, *11*, 2232. [\[CrossRef\]](#)
72. Noardo, F.; Wu, T.; Otori, K.A.; Krijnen, T.; Stoter, J. IFC models for semi-automating common planning checks for building permits. *Autom. Constr.* **2022**, *134*, 104097. [\[CrossRef\]](#)
73. Koo, B.; Shin, B. Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure Building Information Models. *J. Comput. Des. Eng.* **2018**, *5*, 391–400. [\[CrossRef\]](#)
74. Wang, N.; Issa, R.R. Ontology-based integration of BIM and GIS for indoor routing. In *Proceedings of the Construction Research Congress 2020: Computer Applications*, Tempe, AZ, USA, 8–10 March 2020; pp. 1010–1019.
75. BuildingSMART. Latest Available Documentation for IFC 4.3.x Specification. Available online: <https://ifc43-docs.standards.buildingsmart.org/> (accessed on 1 March 2023).
76. Aramali, V.; Sanboskani, H.; Gibson Jr, G.E.; El Asmar, M.; Cho, N. Forward-looking state-of-the-art review on earned value management systems: The disconnect between academia and industry. *J. Manag. Eng.* **2022**, *38*, 03122001. [\[CrossRef\]](#)
77. Open IFC Viewer. Available online: <https://openifcviewer.com/> (accessed on 1 March 2023).
78. Li, H.; Zhang, J. Interoperability between BIM and BEM using IFC. In *Computing in Civil Engineering 2021*; American Society of Civil Engineers: Reston, VA, USA, 2021; pp. 630–637.
79. Khoshelham, K.; Vilariño, L.D.; Peter, M.; Kang, Z.; Acharya, D. The ISPRS benchmark on indoor modelling. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, W7. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.