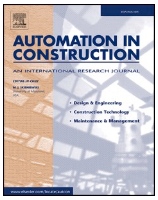
Smart component-oriented method of construction robot coordination for prefabricated housing

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Smart component-oriented method of construction robot coordination for prefabricated housing

Aiyu Zhu [*], Pieter Pauwels, Bauke De Vries

Faculty of Built Environment, Eindhoven University of Technology, the Netherlands

ARTICLE INFO

Keywords:
Smart component
Construction robot coordination
Prefabricated construction

ABSTRACT

Although achievements have been made in research on robotic construction for specific construction tasks, robots are still not capable of working together to accomplish multiple construction tasks. To achieve this goal it is necessary to study how to realize robot coordination in prefabricated construction. In this paper, we propose a component-oriented robot construction approach. Using the smart construction object (SCO) approach, diverse construction tasks are assigned to robots by assigning state and requirements to the components to drive multiple robots for the assembly of prefabricated housing. Within a prototype BIM simulation environment, we implemented multiple different robots to complete the construction of a steel frame based on the SCOs. For more realistic robot-base construction design, the next step is the introduction of more complicated BIM models and more accurate robot models to enable collaborative simulation of a wider variety of prefabricated construction processes.

1. Introduction

In recent years, an increasing amount research on robotic construction shows various advantages such as improving safety, reduction of construction time, and higher quality control [1,2]. However, the adoption of construction robots has been very limited. As a result, the expected benefits are not reached, which is problematic in a time with a demand for mass-scale construction and renovation (e.g. prefabricated housing production - PHP) with high levels of efficiency and quality. Several studies research this phenomenon with various respects, such as limitations of robotic technologies, commercial risks, and constraining of construction [3–5].

1.1. Lacking coordination method

One of the reasons mentioned for the above problem is the lack of a coordination method for construction robots to achieve on-site automation. While robotics (e.g., robotic arms, 3D printing, etc.) are increasingly being used in the construction process, current research is focused on solving isolated construction tasks using single robots. In the construction of a prefabricated house, each construction task is regarded as an independent event.

For example, Bock and Linner [6] proposed the Single-Task Construction Robots (STCRs) which can assist workers in achieving one specific construction task. Depending on the function, STCRs are defined according to 24 categories containing 200 independent systems. Among them, the relevant ones for site prefabrication include: mobile robots, site logistics robots, robots for positioning of components, and robots for building structure assembly. However, each task is part of a continuous construction process. Therefore, when introducing a large number of robots with different functions into the construction process, it is necessary not only to consider the distribution of robots that perform independent construction tasks, but also to coordinate multiple single robots that can complete different construction tasks.

1.2. Automated construction for prefabricated housing production (PHP)

In the context of construction 4.0 [7], prefabricated housing production (PHP) has been proposed as an innovative construction method to promote the efficiency of construction by integrating design, prefabrication, and construction and to achieve full management and automation of the prefabricated construction process [8]. As the two main methods of prefabrication, on-site prefabrication is the assembly of components on the site, while off-site prefabrication is constituted from factory assembly of components to form rooms or housing units, which are assembled on the site. In any case, the scale of construction is far

[*] Corresponding author.
E-mail addresses: a.zhu@tue.nl (A. Zhu), p.pauwels@tue.nl (P. Pauwels), b.d.vries@tue.nl (B. De Vries).

https://doi.org/10.1016/j.autcon.2021.103778
Received 28 January 2021; Received in revised form 16 April 2021; Accepted 19 May 2021
Available online 15 June 2021

bigger than that of traditional manufacturing: both components as well as construction projects are often larger in size. As a result, construction automation can hardly be done in a stationary space like most manufacturing, and we still need on-site work. To address this issue, Bock and Linner [6] introduce On-Site Manufacturing (ONM), which is defined by them as the process of low-level to high-level assembly of components on a site by using fixed or mobile automated machines/robots, rather than the traditional construction process. They claim that the adoption of On-Site Manufacturing (ONM) can potentially achieve large-scale automation of construction processes.

1.3. Robot categories for automated construction sites

Collaborative robotic manipulation is commonly categorized into fixed manipulators, mobile robots and mobile manipulators [9].

A fixed manipulator is a manipulator in which the workstation is fixed. In the manufacturing industry, the use of fixed manipulators has been an unprecedented success that benefits from well accommodated workshops and orderly production lines [10]. The production of sophisticated products can be robust and rapid by establishing a standard assembly line with robots working independently.

A mobile robot is a robot that has the capability of moving and navigating through its surroundings. Mobile robots also significantly improve the efficiency of the logistics industry in warehousing management [11]. Yet, both fixed manipulators and mobile robots are less fit for the large-size, difficult to predict and very dynamic environment that exists in construction industry. These large-size construction environments make it difficult to achieve an entire construction process on an assembly line or in a warehouse.

The third type of robot, namely the mobile manipulators, which are robots that combine a manipulator with a mobile platform for mobile operation, could therefore be considered as the primary choice of robots used in the construction processes considered here. Because of their manipulator function, they are excellent candidates to automate the many actions that need to be done on a construction site.

1.4. The necessity of construction robots coordination

Despite many efforts, it is still challenging to complete an entirely autonomous construction process on-site through construction robots, with regard to both transport and manipulator functions. Davila Delgado et al. [4] has conducted a systematic study on the challenges of robots in construction. In addition to high implementation costs, immature robot technology and other factors, the lack of reliable robots' control and coordination methods for construction robots during the construction process is one of the most significant challenges to realize robot construction.

Although it is obvious that control and coordination methods for industrial robots have been widely used in manufacturing, they are difficult to implement directly in construction. One of the significant differences between the construction industry and manufacturing is that the complex and open construction environment makes the construction process dynamic. Buchli et al. [3] mentioned that the construction process is incredibly complex, so the construction robots need scalable controls.

In other words, industrial robots in manufacturing execute tasks at fixed locations or in less variable environments, while the coordination of multiple robots in construction (Multiple Robot Coordination - MRC) needs to support the construction tasks under varying construction circumstances in different projects. Hence, in the case of On-Site Manufacturing (ONM), there is a gap between this MRC planning and construction (site) management. In this paper, we assume that bridging this gap will lead to a more automated construction site, in which multiple coordinated robots work together to perform their tasks.

1.5. Bridging robot planning (e.g. MRC) and construction management

So far, numerous studies have included methods for multi-robot coordination (MRC). As a critical element of MRC, task planning is used to figure out how to deploy tasks to robots. This task planning is divided into two aspects: task decomposition (TD) and task allocation (TA) [12]. TD relates to breaking down an entire project into several subtasks that can be achieved by different robots with their functions [13]. The definition of TD is similar to the work breakdown structure (WBS), which decomposes tasks into manageable pieces that are called Construction Work Package (CWP) [14]. TA relates to the allocation of workers to the tasks in this construction process.

Task allocation (TA) can be considered as a typical NP-hard problem. Traditional construction management mostly depends on construction managers who heuristically create and adjust the on-site construction processes based on their experience [15]. If the construction managers need also to take robots into account, then the TD and TA planning remain highly needed, yet they need to include robots to perform specific tasks, and task decomposition will change depending on robot capabilities. Hence, it is necessary to extend the construction planning and include the Multi-Robot Task Planning (MRTP). As such, it is possible to create optimal planning of working processes that include tasks for robots [16].

With the booming development and adoption of Building Information Modeling (BIM) technology, construction planning simulation can be performed with more lower-level details. In fact, these BIM models provide an excellent product breakdown structure, which can form the basis of planning based on work breakdown structure. Many studies aim at the creation of a construction planning based on BIM models. For instance, Jeong et al. [15] presented a Just-in-Time (JIT) based lean construction method that integrates BIM data. Lu [17] introduced Discrete Event Simulation (DES) in combination with BIM models as a framework to achieve automation of resource allocations (similar to TA). Furthermore, the introduction of smart construction objects (SCOs) [18] allows construction information to be directly embedded into the physical components (IoT for construction elements, RFID tags, etc.), which potentially makes construction tasks more flexible and independent (lean construction). Further, by integrating SCOs and Construction Work Packages (CWPs), Li [19] proposed the Smart Work Packaging (SWP), which implements constraint management in prefabricated housing production (PHP).

Most of today's multi-robot TA (MRTA) methods are based on the Contract Net Protocol (CNP) model, which is a negotiation system for multi-agents systems [20]. For prefabrication, the number of components and their final target positions are constrained, thus in construction task allocation, this is a deterministic problem without target exploration. However, it is necessary to consider not only the efficiency of the robot to execute each task but also whether the execution of each task can improve the overall construction efficiency. One of the MRTA methods, namely the market-based method, is a method of allocating tasks that mimic bidding in the marketplace to establish the balancing of costs and revenues of individuals and teams to achieve competitive bidding [21], and it can therefore be considered as an appropriate framework to adopt MRTA in construction. It calculates not only the cost of tasks done by robots, but also the revenue of these tasks.

1.6. A component-oriented method for construction robot coordination

Based on the above, we aim in our research to bridge between multi-robot coordination (MRC) and the available research on construction planning, in particular BIM and lean construction, and thereby to achieve a TD method in construction. We aim to rely on the market-based TA method in our proposed MRC method. This will allow us to find a balance between the efficiency of robots and the requirements in construction processes.

This study aims to develop a component-oriented multi-robot coordination method for PHP. In this method, the component is used to bridge between construction tasks and robots. The tasks which are decomposed from work packages are assigned to components (mapping between work breakdown structure and product breakdown structure), and each component is treated as an independent task.

1.7. Outline

In this article, we first perform a literature review (Section 2). In Section 3, we present specific methods on how to define the state and requirements of a smart construction object (SCO) and establish the interaction between the components and the robots to achieve multi-robot coordination (MRC). In Section 4, we will show how this method can be implemented in combination with BIM models using a simple steel structure (a frame with four foundations, four columns and four beams) as an example. Section 5 shows a simulation process for coordination of construction robots by using Section 4's BIM model and robot model. Finally, Section 6 concludes the article with an outlook on future research.

2. Literature review

The study of construction robot coordination covers backgrounds in different disciplines, so in the literature review, we will review each of the three aspects of construction robots, construction planning and robot coordination methods, and explore the feasibility of collaborative robotic construction by using the robot task allocation approach for component-based construction tasks.

2.1. Multi-robot prefabrication

The integrated site concept for Robot-Oriented Design (ROD) has been first proposed in the 1980s [22]. There are two main approaches, namely: (1) an approach based on cooperating STCRs (Single Task Construction Robots), often mimicking human construction processes, and (2) a factory approach, in which a machine-based manufacturing process is targeted, on-site or off-site. Bock investigated the approach based on cooperating STCRs and classified the STCRs in 24 categories for different specific tasks [23]. Hence, the concept emerged of cooperating STCRs in which various STCRs are envisioned to work in collaboration and on-site. The factory approach categorizes 13 systems as an advanced framework for multi-robot construction to achieve automated construction processes as a manufacturing process [24]. Further, the Robot-Oriented Design (ROD) was declared as a future ubiquitous construction method and building technology in the whole Architecture, Engineering and Construction (AEC) field [25]. In the remainder of this article, we will focus on on-site collaborative robots primarily, thereby bringing the Robot-Oriented Design (ROD) as a construction method to the construction site.

In recent years, there has been increasing research focus on using collaborative robots (cobots) for structural assembly. For example, several studies developed multi-agent assembly systems for truss structures. Nigl et al. [26] presented a 3D-printed node-and-strut construction system based on the concept of a climbing robot. Another approach was put forward as Shady3D, which aims at robots gripping the end of components and positioning the component using a single-axis point for assembly with other components [27]. Kuffner et al. [28] designed an automatic robot path planning method based on Rapidly-exploring Random Trees (RRT) which can be used in a lab environment to complete the assembly task by the cobot with discrete elements [29]. Moreover, modular assembly is also a prevalent approach for cobot assembly. The components in the modular assembly system are designed or pre-assembled to make the components apt to connection by robots. For example, the Automatic Modular Assembly System (AMAS) uses inchworm type robots to grab specific bricks and then connect them with other bricks [30,31]. Similarly, Jenett et al. [32] states that a material-robot system with mobile robots can assemble cuboctahedral voxels to form various structures. The TERMES system offers a cooperating working method based on swarm robotics to build customized foams [33]. An alternative method of cobot structural assembly is a self-configuring method that treats robots as structural components to be built by themselves. A configurable system with unmanned aerial vehicles (UAV) is proposed and used for robotic fabrication and assembly [34]. However, such studies remain narrow in focus dealing only with one type of cobots to assemble one type of component. These initial studies need to be scaled up to arrive at multiple types of robots that can work together, also on more complex projects. It is therefore necessary to develop a method that can deploy various types of robots to achieve different structural requirements.

Nevertheless, few studies presented a framework for fabrication by different robots. The 'Hephaestus' is a project aiming to develop a cable robot system for curtain walls' building, repair and maintenance. In this project, multiple types of robots are used in the assembly process [35]. Additionally, the concept of Multi-Machine Fabrication has been proposed, which is an integrative design process by utilizing a combination of drones and industrial robots for fabrication [36]. Furthermore, the in-situ fabricator (IF) as a single robot can execute multiple tasks during construction thereby also achieving multi-robot construction [3]. Considering all of these studies, the vast majority of current research on construction robots focuses on a single robot (with a few achieving more than two) to achieve a specific construction task. These studies have achieved results, but they are relatively fragmented. Even though some construction robots have been used in experimental applications, it is difficult to generalize and combine these different functional construction robots.

2.2. BIM-enabled construction planning

With recent developments in BIM and IoT, it is of particular interest to introduce digital and intelligent technologies into AEC to realize smart construction [37]. Consequently, a lot of studies are focused on how to integrate BIM and IoT into traditional construction project management, especially for prefabrication.

To date, several studies have investigated integrating BIM and project management methods (e.g., Work Package (WP), Discrete Event Simulation (DES)) for construction decision-making. The first comprehensive discussions and analyses of the advanced Construction Work Package (CWP) emerged with Hamdi [14]. In her thesis, three types of advanced work package are defined as Construction Work Package (CWP), Engineering Work Package (EWP) and Installation Work Package (IWP) to describe the manageable pieces in the construction process. To improve the work packaging process accurately and efficiently, the integration of BIM data and mapping matrices are needed to generate the project solution at an early stage [38]. In the same vein, a fluent BIM data flow with the WP is needed to figure out the lacking processes in a Resource-Constrained Project Scheduling Problem (RCPSP) [39]. By integrating BIM and Prefabrication Housing Production (PHP), Li [8,40] presented a framework to optimize decision-making efficiency and collaboration for PHP, which also contains the concept of WP and lean construction. Lu and Olofsson [17] developed an integrated framework based on Discrete Event Simulation (DES) and BIM aiming at decision support in the construction process. Thus, BIM-enabled construction provides support for the application of fine-grained construction planning.

With the development of cyber-physical technology, everything can be treated as an object with computational intelligence. Taking advantage from the new technologies, the concept of Smart Construction Objects (SCOs) was defined as a construction component with three properties: Awareness, Communication and Autonomy [18]. Using SCOs, researchers have been able to explore project management in detail based upon component-level information. Utilizing SCOs, Niu [41,42] respectively proposed management methods for construction logistics and construction equipment. Through the integration of SCOs and advanced WPs, the framework proposed by Li [43] establishes a proactive constraints-modeling method for PHP processes. All studies reviewed show that with the development of BIM technology, the richness of data has led to the study of construction planning on a smaller scale and the description of the construction process is becoming more concrete. Simultaneously, with the integration of prefabricated construction and IoT technology, the concept of SCOs has been proposed to provide ideas to make construction components carry BIM information. Based on BIM-enabled construction planning and SCOs, it is possible to realize independent components with interactive construction task information.

2.3. *Multi-robot task allocation (MRTA)*

A literature review was executed on construction planning based MRTA methods for Single Task Construction Robots (STCRs). In 1980, a seminal article was published entitled 'The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver' which describes the fundament for most MRTA methods. The Contract Net Protocol (CNP) [20] has been proposed to achieve a distributed task assignment method by using a negotiation strategy in multi-agent systems. Based on the Contract Net Protocol (CNP), two main approaches have been developed to solve the allocation under different conditions: the market-based approach and the auction-based approach [12].

The first architecture based on a market-based approach was proposed by Stentz and Dias [21] employing a method that is founded on the market mechanism to realize multi-robot coordination for a given target. By defining the revenue and cost functions, this architecture can perform specified tasks in a series of possible plans. A task is decomposed into multiple subtasks, allowing robots to bid and negotiate to perform these subtasks. Meanwhile, facing one common objective, the robots can cooperate or compete for their own benefits. In order to achieve the execution of the complex tasks, Zlot [44] presents a method using a Boolean logic operator to describe subtasks which are decomposed from the complex task and allows tasks to trade in a market. Also based on the market-based approach, an aerial-ground robotic system has been developed to achieve autonomous delivery through the hybrid vehicle [45]. Furthermore, to enable cobots to work under an unknown dynamic environment, a system called MURDOCH was proposed, which is an online assignment architecture based on an auction-based approach [46]. Compared to the auction-based method, the market-based method can account for cost and benefit but also allows for task reassignment [12].

According to this very brief review of MRTA methods, a precise task description is one of the most critical factors in achieving multi-robot coordination. Therefore, the widely used traditional construction planning methods, which use days or floor completion as task units and are adjusted by the project manager on site, cannot define specific construction tasks for robots. However, in combination with the study in Section 2.2, we have the potential to achieve a construction task that is available to construction robots by using the components as the basic units of the task, combined with BIM-enabled construction planning for a fine-grained construction process.

In this paper, the traditional MRTA method will be used to develop a new cobot-oriented construction planning method. Other derivative methods based on market-based approach and auction-based approach [47–49] are not included and left to target for the further research.

3. Methodology

Unlike many manufacturing industries, construction processes mostly occur in an open environment (on-site construction). Moreover, the high freedom of architectural design makes it difficult for site construction to have a fixed manufacturing process like an assembly line. Therefore, there exists high uncertainty and a need for flexibility in the construction process, which makes planning of the construction process challenging to be accurate for every step in the process. In order to realize multi-robot construction, specific construction tasks and methods must be clearly defined. Consequently, an approach is needed for each step of the construction process with well-defined components and tasks. Based on these definitions, a component-based construction method can be developed for multi-robot-coordination.

We refer here to a new method that is different from the component-based method used for prefabrication and Smart Work Packaging (SWP). In particular, we suggest to assign the information of construction tasks to the components in the construction process as discrete events. As such, these components become SCOs with awareness, communication, and autonomy in terms of how they need to be built by robots. The construction task of each independent component can be assigned to the STCR that meets the construction requirements of the components through the market-based method (MRTA), to realize the collaboration of multiple STCRs. We want to use only low-level rules (component's states and requirements) to describe a high-level construction process.

The method consists of three parts:

1. Definition of attributes of SCOs;
2. Construction TD for the construction of SCOs;
3. Introduction of the Construction TDs into a Multi-Robot Task Allocation (MRTA) process.

This method uses BIM models as input data. In particular, Industry Foundation Classes (IFC) files are used to deliver and describe the information of the components. Based on the IFC data, robots are assigned tasks (MRTA), as indicated in the above three steps. In the following subsections, the three individual parts of the above method are explained in more detail, followed by a section brings with a show case from practice.

3.1. *Definition of attributes of SCOs*

The concept of Smart Objects (SOs) derives from the concept of ubiquitous computing. In 2016, Niu [18] expanded the idea of Smart Objects into Smart Construction Objects (SCOs). As the fundamental construction blocks for future construction, the concept of SCOs defines that the resources (e.g., machinery, devices, and materials) in construction can be given intelligence as objects [18]. Thus, each SCO can have independent information and different functions. By combining with IoT technology, SCOs enable data interaction between different objects in the construction process. In several prefabricated construction studies, SCOs are applied in construction Logistics and Supply-Chain Management (LSCM), safety management, facilities management, etc. In the study of the SCOs-enabled construction process, prefabricated components are considered as SCOs and installed with smart hardware (e.g., microcontroller, RFID) to track the components' spatial and timing information during the construction processes, thus achieving lean construction and improving the overall construction efficiency [8,41,50]. The concept of SCOs as a future construction method needs more applications to be tested and refined; however, its reliability is demonstrated in the current application in specific projects and simulation environments.In our approach, the components have to carry their information, interact with the construction robot for their construction tasks, and alter their state according to the construction requirements. The three core properties of SCOs enable the definition of the components in our methods.

Three core properties of SCOs are proposed: Awareness, Autonomy and Communication. This section describes how to define such SCOs in the case of structural constructions. We hereby distinguish specifically between Component and their ConnectionNodes and inherit attribute from construction object (see partial UML Class Diagram in Fig. 1).

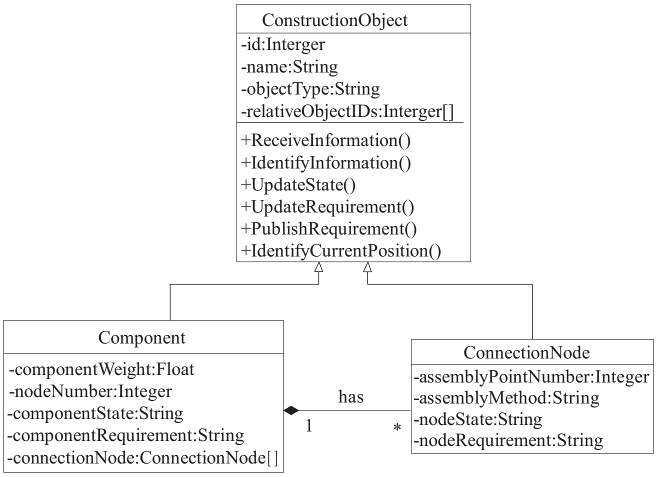


Fig. 1. UML Class Diagram for a SCO.

3.1.1. *Awareness*

The awareness of components refers to the real-time state of the components and their perception of the environment. These can be defined in different ways. As mentioned in Section 3.2, the construction process is considered in this paper as a sequence of discrete events. Even if there is a correlation between components, the tasks of each component are independent. Therefore, in addition to the definition of SCO awareness (such as ID, Name, ObjectType, RelativeObjectIDs, etc.) in Fig. 1, we assign all construction tasks related to the component as awareness. The breakdown of construction tasks is detailed in Section 3.2.2.

For example, a column (Component) is connected to other components at the foot (ConnectionNode) and the top (ConnectionNode) of the column. The foot of the column has six (AssemblyPointNumber: six) bolt anchor (AssemblyMethod:bolt) nodes to the foundation, and the top of the column is connected to two beams, each of which also has six bolt anchor nodes. Therefore, the awareness of the column includes Component and ConnectionNode, both of which are subclasses of a common ConstructionObject class. The properties of the ConstructionObject include:

- The ID, Name, and ObjectType from the BIM model,
- The RelativeObjectIDs records the IDs of the relative components (foundation and the two beams in this example)

For each ConstructionObject, we furthermore assume the availability of spatial information and shape information. This spatial information is divided into two parts: the current position of the column during construction and the position of the column after it is built from the BIM model. This shape information is used to represent the shape of the component and this data will load from the BIM model. The information is kept out of the UML class diagram in Fig. 1.

The properties in Component include:

- The ComponentWeight from BIM model,
- The NodeNumber represents how many connection nodes the component contains (three in this example),
- The ComponentState (e.g, InTransit, WorkingInPosition, etc., see Section 3.2.2 in details) is the current construction state of the component.
- The ComponentRequirement (e.g, NeedToTransfer, NeedToPosition, etc., see Section 3.2.2 in details) is the construction requirement under the current componentState.
- For each Component, a number of ConnectionNodes can be assigned, and they are stored in an array of ConnectionNode objects.

Futhermore, each ConnectionNode has the following properties:

- The AssemblyPointNumber indicates how many nodes a ConnectionNode has to connect to (six for this example),
- The AssemblyMethod indicates what method is used to assemble (bolt in this example).
- The NodeState (e.g, InLocation, WorkInPosition, etc., see Section 3.2.2 in details) is used to describe the state of the ConnectionNode under current componentState.
- The NodeRequirement (e.g, NeedToPosition, NeedToAssembly, etc., see Section 3.2.2 in details) is the construction requirement of the component's connection node under the current NodeState.

The functions in ConstructionObject are used to interact with information, update state and determine the current position of the Component and ConnectionNode.

3.1.2. *Communication*

Communication is the ability to output information based on awareness. The output in this method is divided into two parts: in the first part, output requirements of SCOs as Communication functions in Fig. 1. We allow Component and ConnectionNode are both inherit functions from construction object, so both of them can release the construction requirement to the robot as a construction task (see Section 3.3); in the second part, the state of the component is used for planning the construction process (see Section 3.2.1).

For example, when an assembly task is needed from the component in the construction area, the preconditions will be determined first. If the assembly component is not on the final location, it will be searched in the yard area as the state of a component is 'OnYard'. At this time, components that meet the requirements will release the requirement 'NeedToTransfer' to the robots.

3.1.3. *Autonomy*

Autonomy is an alternative approach to SCO communication that allows components to act autonomously according to predefined rules. In this case, the component is the target object of both the construction tasks and the robot tasks, so the consistency between the construction task and the robot task should be guaranteed, and the autonomy of the component should be constrained and maintained. Component autonomy under construction implies the change of the state of the component when a robot realizes construction requirements of the component.

For example, when a robot from the yard loads a component, the transport task starts. The state of the component should be updated after the robot action is completed. The state of components changes from the state 'OnYard' to state 'InTransit', and the requirement of components changes from the state 'NeedToTransfer' to the state 'NeedToPosition'.

3.2. *Construction TD for the construction of SCOs*

The second part of the method is the construction task decomposition for SCOs. We distinguish hereby the overall policy for component-oriented construction (Section 3.2.1) and the more detailed breakdown method for construction tasks (Section 3.2.2).

3.2.1. *Component-oriented construction regulation*

Different from cast-in-place construction, prefabricated construction relies mainly on the assembly of components. Components are prefabricated and assembled on-site. Therefore, the construction method in this case is the process of moving components from the yard area to the position requiring assembly in the structure and assembling them. In order for the robot to perform such a construction process, we need to implement a breakdown of the process, which is transformed into a sequence of actions that robots can execute.

In this study, we follow the theory of SCOs with main characteristics: Awareness, Communication and Autonomy - [18]. Among others, the component's spatial information includes two parts, the initial placement and the target placement. Since the BIM model has already been completed, we can directly access the component's spatial position in the completed structure, taken as the target placement. To simulate the construction process, we assume that the component has not started to be built, which is the component's initial state: 'OnYard'. The construction process is to transport the components from the initial placement to the target placement and accomplish the assembly, while the component state changes from an initial state ('OnYard') to a final state ('Assembled'). Such awareness includes the position of components in space, representation of components, connection relationship between components, and state of components. By comparing the component's awareness in the construction process with the component's awareness under the completed structure, it is possible to update the component's status and requirements, and to publish a construction task to the robot. Besides awareness, the SCO also possesses autonomy and communication (see Section 3.1 for more detail). As a result, the SCO component can also propose new construction requirements according to the state changes in the tasks, in order to realize a dynamic construction based on SCO. All proposed construction processes will be aggregated into a process that satisfies the requirements of all components.

The construction planning based on SCOs is organised as follows:

1. Traverse the components on the construction site which have published their requirements for a construction task.
2. Match component requirements to the current construction tasks.
3. Publish the components' requirements corresponding to the construction tasks.

3.2.2. Breakdown the construction tasks for SCOs

In the above procedure, construction tasks are broken down into smaller tasks, and all components have relationships with other components. We treat components as the targets for the construction task breakdown. Components are associated with construction tasks. Construction tasks are assigned to components with different properties through the SCO method. In our method, components publish tasks to be executed by robots. The construction process in the case of prefabrication and on-site assembly can be summarized as: 1. Transport task: transfer the components from the yard to the structure; 2. Positioning task: Locate the components' positions; 3. Assembly task: Assemble the components.

Since the individual construction tasks are assigned to a component, two properties are needed for every component and their nodes: 1. the states of the Component and ConnectionNode are used to indicate the current state of the component and its connection nodes, 2. requirements of the Component and ConnectionNode are used to represent the construction tasks required for the component and its connection nodes in their current state. In the case of on-site assembly, we can distinguish three construction tasks with properties. These three construction tasks are displayed diagrammatically in a UML State chart diagram in Fig. 2. The transport task aims to transfer the component from the storage yard to the final location on the construction site. Therefore, during the transport, the possible states of a component are: 'OnYard', 'InTransit' and 'Arrived'; and the requirement of a component is: 'NeedToTransfer'. Once the component has arrived to the right location, the component will enter the positioning task. In the positioning task, the component and its nodes need to wait in the arrived position while the positioning robot has not reached the working position. When the positioning robot reaches the position and starts positioning work, the component enters the 'WorkingInPositioning' state. When the positioning is complete, the positioning task of the component is finished. So, the possible states of a component and its nodes are: 'InLocation', 'WorkingInPositioning', 'PositionFixed'; and the requirement of a component and its nodes is 'NeedToPosition'. When the

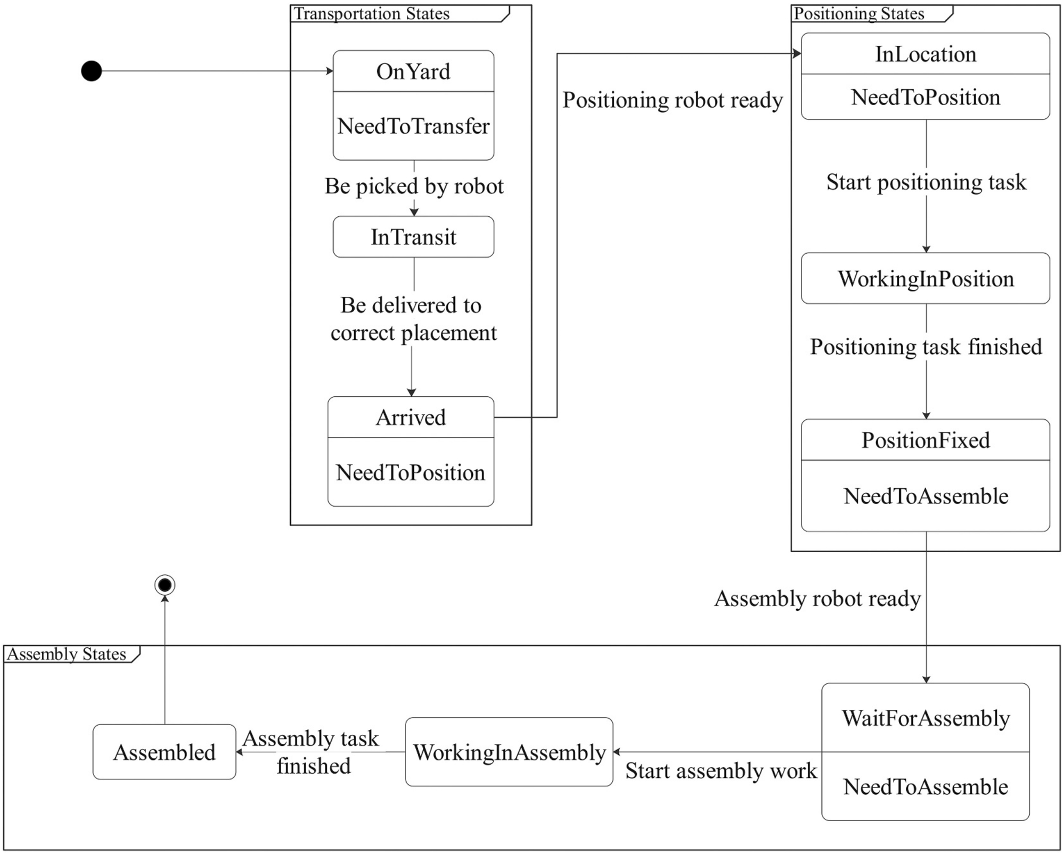


Fig. 2. Statechart diagram for the smart construction object.

position is fixed, the assembly work starts. The assembly task aims to assemble the connected components into a structure. In the assembly task, the component have to wait for the assembly robot to reach the correct construction position, after which the assembly work is executed and completed. When the assembly task is completed, the construction task for the component is finished and the component's state is assembled and there are no further requirements. So, in an assembly task, the states of the component and its nodes are: 'WaitForAssembly', 'WorkingInAssembly', 'Assembled'; the requirements of the component and its nodes is: 'NeedToAssemble'.

The state and requirements of the node should be consistent with the components, but in assembly state there are sometimes differences. For example, when construction is performed on a column and the foot of the column is already connected to the foundation, the state of the foot nodeState is 'Assembled' and there is no construction requirement; however, the top of the column is not yet connected to the beam and the nodeState is 'WaitForAssembly' and nodeRequirement is'NeedToAssemble'. At this time, componentRequirement is still 'WaitForAssembly',componentRequirement is 'NeedToAssemble', only when all ConnectionNodes are 'Assembled', the componentState changes to 'Assembled'.

3.3. Introduction of the construction TD into a MRTA process

Stentz and Dias [21] proposed a market-based approach for task allocation (TA) in 1999 (MRTA), which defines possible construction plans with cost functions for each task. This method divides a continuous task into subtasks and allows the robot to bid and negotiate to perform these subtasks. The objective is to optimize task allocation by using individual robots that cooperate and compete. Different from other robot task allocation, the market-based approach takes not only the work cost of the robot into account but also the benefit from the task (reward). In construction processes, not only the work cost of the robots but also the time cost should be considered. To achieve this, task allocation for the construction robot can be used based on the market-based method.

In our approach, we treat the SCOs' requirements as specified tasks and treat each construction task that is needed to assemble the component to become part of the final structure as a subtask. For this purpose, we adjust the definitions of revenue and cost from the market-based approach. The execution time of the construction task is treated as the cost of the task, and the operation time of the robot actions is treated as the cost of the robot. When the two costs for the whole construction process add up to a minimum value, robot task allocation with the shortest construction time is found.

In Section 3.2.2, the construction process is defined as three tasks: transport, positioning and assembly. This allows the actions of the robots to be allocated according to these three tasks. In a transport task, the requirement of the component only consists of a single task. When the positioning task and assembly task are performed, construction tasks of the component are further divided into the construction tasks of each connection node, and the construction requirements of each connection node are released as independent tasks. The requirements of every component on the construction site are published as the tasks. The robots that can complete the tasks of these components are matched using the market-based method to execute the tasks. While a component is in progress, the task is terminated unless a state of emergency (collision, etc.) occurs. A component cannot change the requirements or state while it is in progress, nor can it publish a new task until the current task is finished. The process of robot task allocation is as follows:

1. Transport task: The creation of transport tasks is always accompanied by the requirement for components. A transport task occurs when one node of a component has already been assembled but the other node needs to be connected to its associated component (e.g. the foot of a column is already connected to the foundation and the top of the column needs to be connected to the beam). Before the transport task is published by the component, it will be determined that the other component to be connected has been delivered to the specified location as a determination process. If it has already been transported to the specified location, it is an arrived connection component (ACC); if it has not yet been delivered to the specified location, it is a required connection component (RCC). If the component is an ACC, the transport task will not occur and the component will issue a'NeedToPosition' demand to match the positioning robot. When a component is detected as RCC, the system searches the yard for the location of the component and publishes the transport requirement for the component to the transport robot. The robots will calculate their respective positions to reach the component and the robot with the lowest bidding price can perform the task.
2. Positioning task: When a component arrives to its final location, the requirement of the component is changed to 'NeedToPosition', which information is published to the positioning robots. The robot that can execute the positioning task will calculate the distance between the position of the component and its own position as the cost of the robot. This information is used to calculate the minimum cost and determine the robot task to become part of the construction execution plan. When entering the positioning task, the component opens up the state, requirements, and location of its connection nodes, which are used to implement the component positioning and assembly tasks.
3. Assembly task: When the component position is fixed, the component publishes the assembly task and the connection method of each node. The assembly robot will first identify the connection method of the node. The robot that matches the required funcionality will calculate the time required according to the distance between its own position and the component's final location. To calculate the total cost of the robot action, the number of nodes which need to be connected are also considered. When all of the component's nodes (ConnectionNode) are 'Assembled', the component state will be updated to connected to indicate that the component's construction tasks are finished.

Fig. 3 displays the full Flowchart diagram for task allocation.

4. The BIM and robot models in simulation

To validate the method, a construction simulation based on a steel frame is developed in a BIM environment. We present first the use case BIM model (Section 4.1), then how component relationships are represented in this BIM model. We show how this data can be transformed from IFC into SCOs (Section 4.2). Finally, we explain the robot model (Section 4.3) and the entire simulation of construction processes (Section 5).

4.1. The use case BIM model

We develop a simple simulation environment that contains a construction area and a steel frame (Fig. 4). The simple model is primarily used to test the component-oriented construction of the robotic coordination, without considering the optimization of the structure, and is only used to realize the application of the existing BIM model. Therefore, we assume that the model is built by a professional structural designer and meets our modeling requirements.

The steel frame consists of two levels: level 1 contains the footings, plates; level 2 contains the beams and clip angles. Level 1 is located at the base of the columns, and level 2 is at the top of the columns. The four steel columns are W250x73, length 4000 mm, weight 289.81 kg. The four steel beams are W310x38.7, span 2846 mm and 3500 mm, weight 105.49 kg and 134.39 kg (Fig. 5). The BIM model of the structure has been designed based on the requirements of the method and it is necessary to declare the connections of the components. The case is

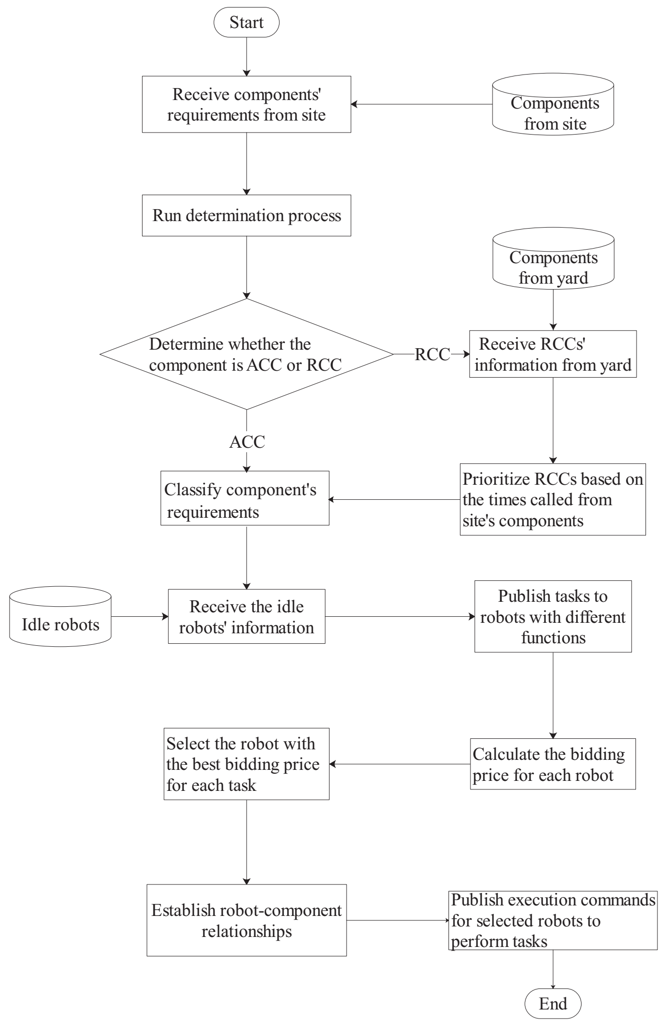

Start
Receive components' requirements from site
Components from site
Run determination process
Components from yard
Determine whether the component is ACC or RCC
RCC
Receive RCCs' information from yard
ACC
Classify component's requirements
Prioritize RCCs based on the times called from site's components
Idle robots
Receive the idle robots' information
Publish tasks to robots with different functions
Select the robot with the best bidding price for each task
Calculate the bidding price for each robot
Establish robot-component relationships
Publish execution commands for selected robots to perform tasks
End

Fig. 3. Flowchart diagram for task allocation.

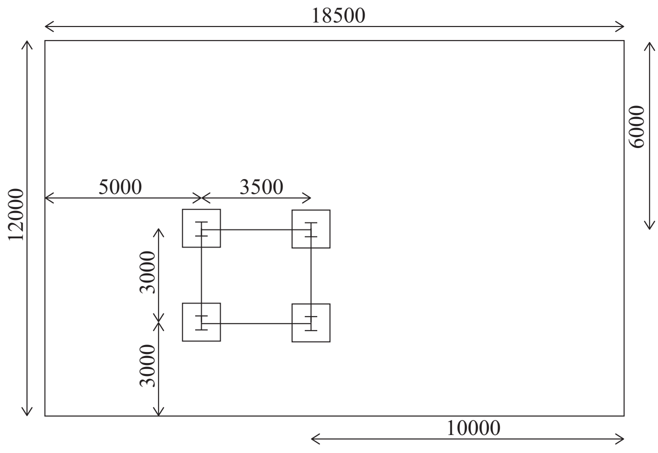


Fig. 4. Layout of construction environment.

modeled using Revit 2021 and contains the site, footings, steel columns, steel beams and the connection methods of components.

In Revit 2021, we use the Connection method of the Steel tool to implement the connection between components. The base plate is bolted to the column and isolated footing. The beams and columns are bolted to each other via clip angles.

We use this model to simulate the process of transporting components from the yard to the placement in structure, and then connecting them to a complete structure by multiple robots working together and generating a sequential construction process. The position of the component in the BIM model is the components' placement in structure, and the initial position of the components in the yard area can be initialized by the user. The specific construction process will be expressed into component requirements using the methods described in Section 3. In this process, the robot will interact with the SCOs in the environment (beams, columns, connections) to obtain the specific construction steps that need to be executed by multiple robots at each step of the construction process. The specific simulation process is described in Section 5.

4.2. Component relationships via IFC

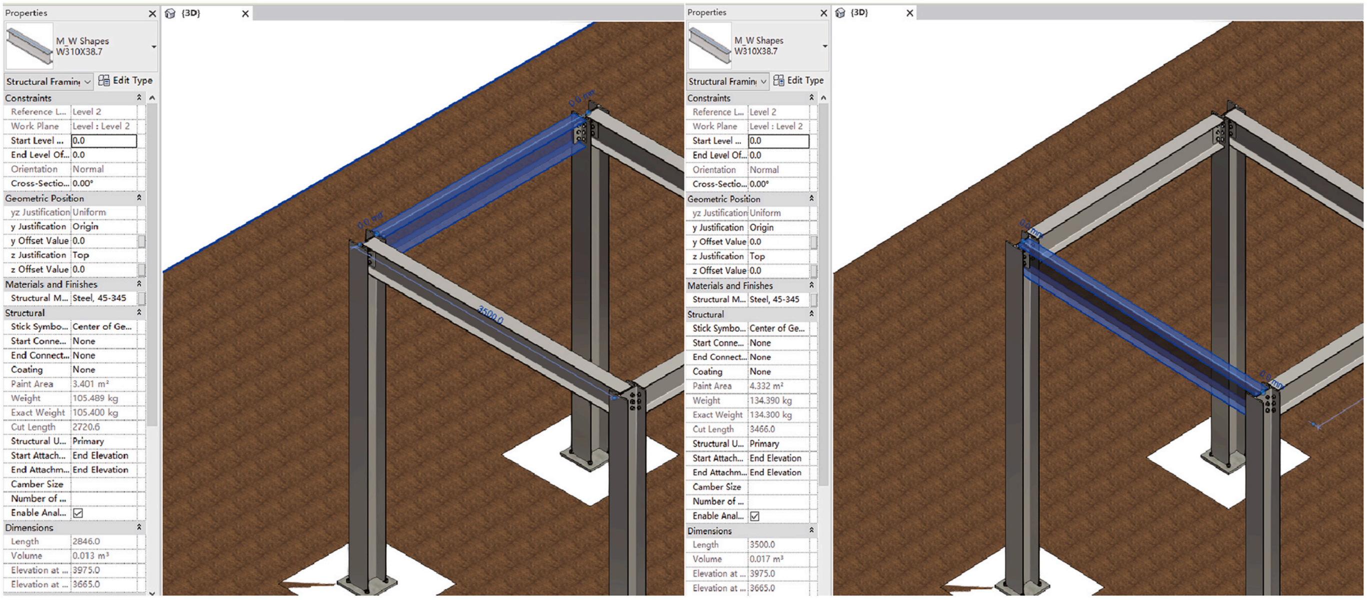
The Industry Foundation Classes (IFC), the international open BIM standard, supports a vendor-neutral representation of BIM models. Hence, we used this format (IFC4) to represent the Revit data. Furthermore, ifcopenshell[1] is an open source software library that can be used to work with the IFC information using python. Hence, we will use Python in combination with IFC to realize the method.

In Section 3.1, we introduced construction-oriented smart construction objects (SCO), which are SCOs that contain a Component object and ConnectionNode objects. The component object is used to describe the properties of the component itself, and the connection node objects are used to describe the connections between the components. In particular, there is some information that we can get from the IFC. For the component object, we can get directly from the IFC: the Component's ID (GlobalID), the Component's name (Name), the Component's type (ObjectType), the componentWeight (Weight), the component's spatial information (PlacementInStructure) and the component's shape information (ShapeRepresentation).

In order to establish ConnectionNode objects and establish SCO-oriented component connection relationships, we need to utilize IfcRelConnectsWithRealizingElements. IfcRelConnectsWithRealizingElements is used in IFC to describe the relationship between two elements (RelatingElement and RelatedElement) in a unified format. Fig. 6 shows the hierarchy of IfcRelConnectsWithRealizingElements in IFC and the attributes it contains. IfcRoot (attributes: 1–4) is the basic information of the connection relationship in IFC format and attributes: 5–9 can be used to establish connection node objects and connection relationships. ConnectionGeometry is used to describe the shape of the connection, RelatingElement is the connected component (e.g., beam, column), RelatedElement is the connector (e.g., plate, clip), RealizingElements are the connecting parts (e.g., bolt), and ConnectionType is the connection method. For the node object, we can get directly from the IfcRelConnectsWithRealizingElements: the number of connection points (the number of RealizingElements), the connection methods (ConnectionType) and the position of the connected node in space (the position of each RealizingElement). Since IfcRelConnectsWithRealizingElements describes a unique set of connection relationships between component (RelatingElement) and connector (RelatedElement), we can generate a new ID of ConnectionNode to identify a unique ConnectionNode object by combining the GlobalID of RelatingElement and the GlobalID of RelatedElement. Furthermore, since components with a connection relationship to each other will use the same connector, we can determine the connection relationship by pairing the same connector.

In our case, the foundation and columns define the connection between the components via plates, and the beams and columns define the connection between the components via angle clips, which are mechanically fastened by bolts. So, for a beam-column node fastened with a bolt via an angle clip, the connection is split into a column and an angle

[1] http://ifcopenshell.org/



Properties

M_W Shapes
W310X38.7

Structural Framin ∨ Edit Type

Constraints	
Reference L...	Level 2
Work Plane	Level : Level 2
Start Level ...	0.0
End Level Of...	0.0
Orientation	Normal
Cross-Sectio...	0.00°
Geometric Position	
yz Justification	Uniform
y Justification	Origin
y Offset Value	0.0
z Justification	Top
z Offset Value	0.0
Materials and Finishes	
Structural M...	Steel, 45-345
Structural	
Stick Symbo...	Center of Ge...
Start Conne...	None
End Connect...	None
Coating	None
Paint Area	3.401 m²
Weight	105.489 kg
Exact Weight	105.400 kg
Cut Length	2720.6
Structural U...	Primary
Start Attach...	End Elevation
End Attachm...	End Elevation
Camber Size	
Number of ...	
Enable Anal...	☑
Dimensions	
Length	2846.0
Volume	0.013 m³
Elevation at ...	3975.0
Elevation at ...	3665.0

Properties

M_W Shapes
W310X38.7

Structural Framin ∨ Edit Type

Constraints	
Reference L...	Level 2
Work Plane	Level : Level 2
Start Level ...	0.0
End Level Of...	0.0
Orientation	Normal
Cross-Sectio...	0.00°
Geometric Position	
yz Justification	Uniform
y Justification	Origin
y Offset Value	0.0
z Justification	Top
z Offset Value	0.0
Materials and Finishes	
Structural M...	Steel, 45-345
Structural	
Stick Symbo...	Center of Ge...
Start Conne...	None
End Connect...	None
Coating	None
Paint Area	4.332 m²
Weight	134.390 kg
Exact Weight	134.300 kg
Cut Length	3466.0
Structural U...	Primary
Start Attach...	End Elevation
End Attachm...	End Elevation
Camber Size	
Number of ...	
Enable Anal...	☑
Dimensions	
Length	3500.0
Volume	0.017 m³
Elevation at ...	3975.0
Elevation at ...	3665.0

Fig. 5. Components' specification in Revit.

```
IFCRELCONNECTSWITHREALIZINGELEMENTS Attributes
IfcRoot: 1.GlobalID, 2.OwnerHistory, 3.Name, 4.Description
 └IfcRelationship
    └IfcRelConnects
       └IfcRelConnectsElements: 5.ConnectionGeometry, 6.RelatingElement, 7.RelatedElement
          └
```

Fig. 6. Attribute inheritance of IfcRelConnectsWithRealizingElements.

clip connection and a beam and an angle clip connection. The column and the angle clip are the related elements and the relating element, respectively, which are connected by bolts. Therefore, we can relate a component to a component by the relationship between the connector (e.g plates, angle clips) and the component. When the relationship of the components is confirmed, it will be determined whether the relationship between the components or between the connected nodes of the components is established. If we are establishing a connection between components, in case of creating component objects, the system will create two SCOs and will assign the properties of IfcRoot in these two RelatingElement to SCOs respectively. Simultaneously, the system will assign the GlobalId of the connected components to each other's RelativeObjectIDs. In case of creating connected node objects of components, two node objects will be created and the IDs of respective RelatingElement and RelatedElement will be merged to generate new IDs to be assigned to node objects respectively, and then the properties in IfcRoot will be assigned to node objects respectively. Finally, the Nodes' ID of the connected node is assigned to the RelativeObjectIDs of the other node. Fig. 7 shows a flow diagram that indicates how this information can be loaded, leading to SCOs with all required information.

4.3. The robots' model

As indicated before in Section 3.3, robot tasks are classified here as transportation, positioning and assembly depending on the type of work they perform. Transportation robots are robots that transfer components from the yard to their placement in the building structure. Positioning robots are robots that help in positioning the component in space to align the points on the nodes of the component to satisfy the connection requirements before performing the assembly. Assembly robots can execute assembly or welding work.

In the robot task allocation problem, the task allocation is based only on the specification of the robot. A robot can only be allocated those tasks that it can perform. The specification of a robot will give different definitions depending on the application scenario. Also, different robot companies will give their own standards. However, there are some basic specifications that are defined both at the design and manufacturing of the robot [51,52]. In conjunction with the requirements of the construction task, we have selected specifications for construction robots that execute diverse tasks. This has led to a simplified UML Class diagram for robots as depicted in Fig. 8.

In the transport process, we want the components to be transported safely and quickly to their intended location, so the specification of the mobile robot should include: payload and robot movement speed. The size of the components needs to be considered, so we should also consider platform size for mobile robot. The freedom of the manipulator, the motion range, the vertical reach and the horizontal reach are also important specifications in order to achieve the transport of the component to any position in the structure.

When a component is transported to a location in the structure, it is necessary to assemble the nodes of the component, which requires a robot with accurate, safe and fast positioning and assembly capabilities. Therefore, we take the above mentioned specifications from real robots to simulate different types of robots. We define the parameter model of the robot according to the construction process. The main parameters of the manipulator are: payload defines the weight of the robot arm so that we can judge the weight of the component to be gripped by the arm; motion range defines the working range of the arm so that we can judge whether the component can be placed in the required space by the arm; handling capacity defines the amount of time the arm needs to complete a job to estimate the time needed to complete a construction task.

The platform size defines the size of the robot's placing area so that one can determine whether the size of the component can be loaded or not. The main parameters of a mobile manipulator are the parameters of the manipulator (above three), plus the parameters of mobile.

Currently, in this case, the functionality of the robot will be abstracted to the robot's performance parameters, regardless of its actual execution methods. When the performance parameters of the robot satisfy the construction requirements of the component, the robot is considered to be able to perform the job. For example, if we need to transport a W310x38.7 steel beam with a weight of 173 kg and a span of 4000 mm, we need the payload parameter of the robot to be greater than the mass of the component and the platform of the transporting robot to be greater than the dimensions of the component. For positioning and

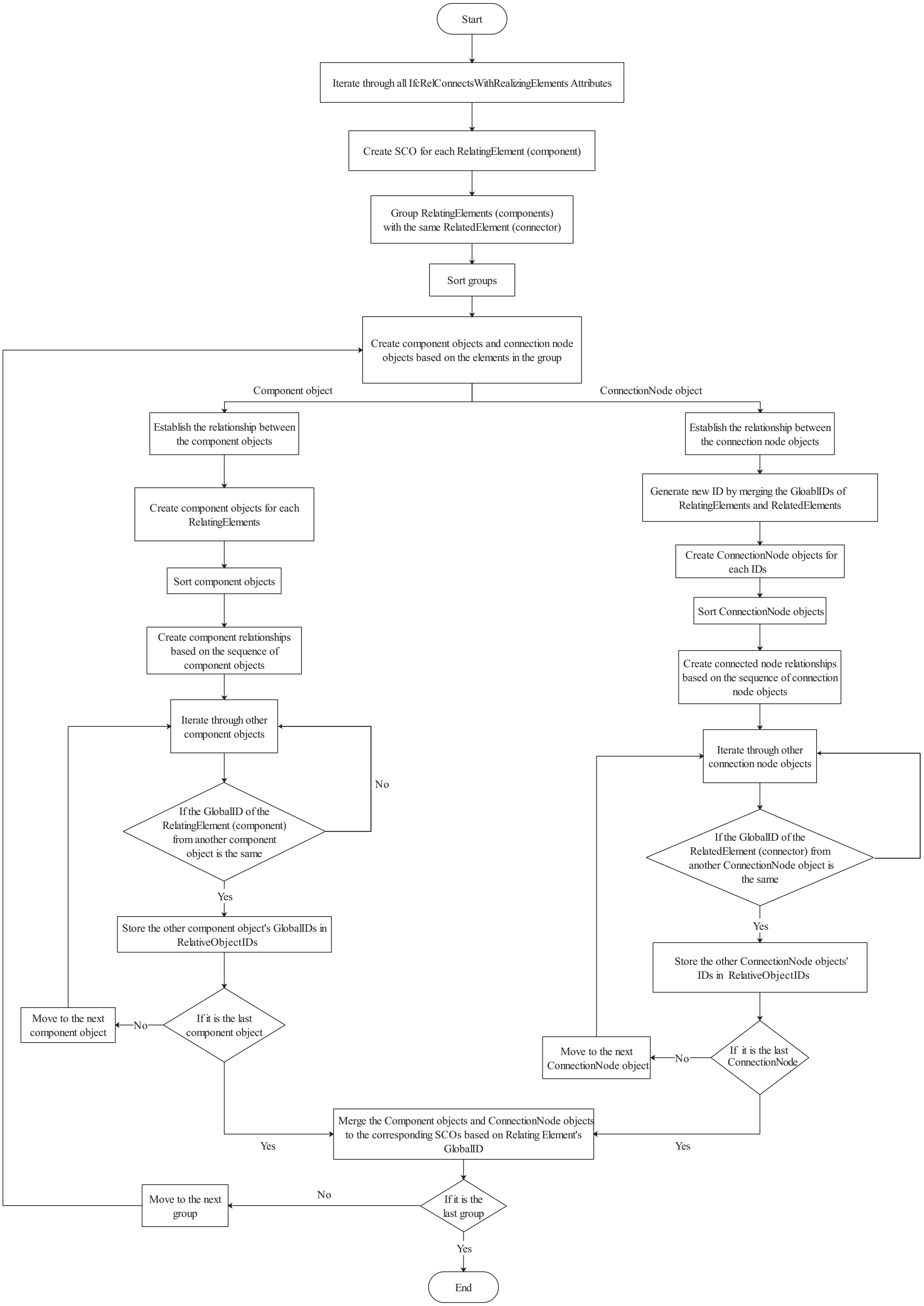

Start
Iterate through all IfcRelConnectsWithRealizingElements Attributes
Create SCO for each RelatingElement (component)
Group RelatingElements (components) with the same RelatedElement (connector)
Sort groups
Create component objects and connection node objects based on the elements in the group
Component object
ConnectionNode object
Establish the relationship between the component objects
Create component objects for each RelatingElements
Sort component objects
Create component relationships based on the sequence of component objects
Iterate through other component objects
If the GlobalID of the RelatingElement (component) from another component object is the same
No
Yes
Store the other component object's GlobalIDs in RelativeObjectIDs
If it is the last component object
No
Move to the next component object
Yes
Establish the relationship between the connection node objects
Generate new ID by merging the GloablIDs of RelatingElements and RelatedElements
Create ConnectionNode objects for each IDs
Sort ConnectionNode objects
Create connected node relationships based on the sequence of connection node objects
Iterate through other connection node objects
If the GlobalID of the RelatedElement (connector) from another ConnectionNode object is the same
No
Yes
Store the other ConnectionNode objects' IDs in RelativeObjectIDs
If it is the last ConnectionNode
No
Move to the next ConnectionNode object
Yes
Merge the Component objects and ConnectionNode objects to the corresponding SCOs based on Relating Element's GlobalID
If it is the last group
No
Move to the next group
Yes
End

Fig. 7. Flowchart for how to convert IfcRelConnectsWithRealizingElements to SCOs.

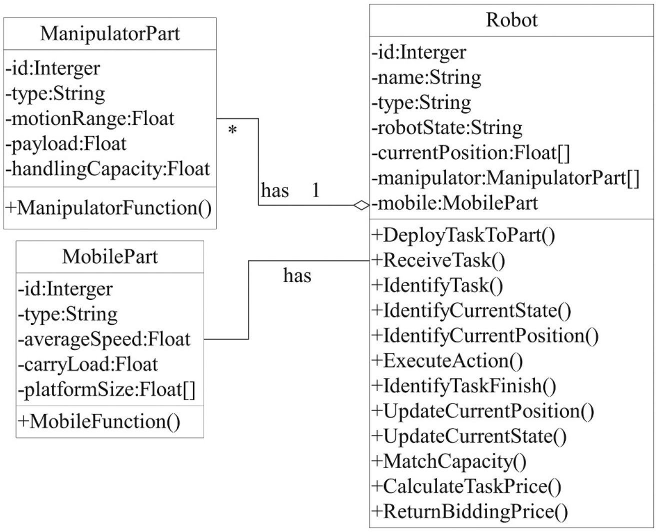


Fig. 8. UML class diagram for robots.

assembly tasks, we require a robot with a payload that is sufficient to perform the gripping and with a manipulator motion range that is sufficient to transport the component to the placement in space.

In order to simulate diverse construction possibilities, the virtual robots listed in Table 1 are defined based on the abstracted robot mobile and manipulator parts and the three performances of each part, with only two correlated constraints for each type of robot. The payload and average speed of the mobile robot are negatively correlated, while the motion range, payload and handling capacity of the manipulator are negatively correlated. We only tested the robot using the parameters in Table 1, which can be customized depending on the performance of the actual robot.

5. Simulation of construction processes

All above components are included in a system that allows simulating construction processes. This system is built in Python, it can import IFC files and generate the SCOs defined in the method, as well as custom-build the robot model according to the given parameters. The simulation process in the system has three phases (Fig. 9). The first phase is the import and generation of the model, the second phase is the initialization of the components and the robot, and the third phase is the simulation of the build process. Fig. 9 displays the full UML Activity diagram for simulation process.

5.1. Phase 1: model generation

At first, we built our test model with Revit and used the connect method in Revit to establish connection relationship to the components. We then exported the file in IFC4 format. Afterwards, we imported the model's IFC file into the python environment via ifcopenshell to facilitate access to the site and component information in our method. The system divided the information from IFC into site information and component information, and converted them to satisfy the requirements of our method:

- The site needs to be initialized and divided into a construction area and a yard area, which happens as part of the Python code. The size of each functional area can be customized according to the project needs.
- Based on the approach in Section 4.2, the system will convert the components in the IFC (e.g., ifcBeam, ifcColumn) into SCOs. In our case, the structural components: four foundations, four beams and four columns, will be converted into SCOs. Connectors such as plates, angle clips, will be defined as connection nodes in SCOs.

5.2. Phase 2: initialization of robots and components

After the SCOs are generated by the system, the initialization phase begins. Initialization requires defining the state and the requirements of the components and the initial state of the robot, as well as their initial positions. Our method currently only simulates the construction process above ground, so the footing is considered to be already built on the site. The initial state of the footings is 'PositionFixed', their requirement is the need for assembly, and their initial positions are the spatial positions of the components in the model.

The initial positions of the unbuilt components (in this case, beams and columns) are at the initial state ('OnYard'), and their requirements are at the state 'NeedToTransfer'. Similarly, the robot state is the initial system-defined state: all robots are idle at the initial state, their initial position is defined by the user, yet within the yard area only. It is also necessary to initialize the construction floor, the number of components to be constructed, and the components that have been constructed. This is used to determine the current construction progress.

In our case, the spatial position of the component in transit is abstracted as the midpoint of the component, and the spatial position of the connection node in positioning and transit is the center of the node point (Fig. 10). The spatial position of the robot is the abstraction of the center of the robot's bottom surface in space.

As shown in Fig. 11, we have initialized the component positions, robot positions and the construction area for the selected case. These parameters need to be customized according to the user's needs (cfr. Other available cases).

5.3. Phase 3: simulation of build process

When the initialization is complete, the construction phase of the simulation begins. A component will be constructed in three stages: transportation, positioning, and assembly. There are three levels of judgment of construction completion. First, for a single component, the construction of the component is complete when all connected nodes are connected. For a single floor, construction is complete when all the connecting nodes are connected except for the one that connects to the vertical structure on the next floor. For the entire structure, when all components are connected, the construction of the entire structure is complete. The flow of construction is based on the sequence of floors,

Table 1
The virtual robots used in this experiment, including their features.

Robot types	Mobile			Manipulator		
	Carry load	Average speed	Platform size	Payload	Motion range	Handling capacity
Transport robot 1	300 kg	0.5 m/s	4 mx1m	NA	NA	NA
Transport robot 2	300 kg	0.8 m/s	4 mx1m	NA	NA	NA
Positioning robot 1	NA	1.0 m/s	NA	300 kg	4 m	120 s/node
Positioning robot 2	NA	1.0 m/s	NA	300 kg	4 m	60/node
Assembly robot 1	NA	1.0 m/s	NA	NA	4 m	1 points/60s
Assembly robot 2	NA	1.0 m/s	NA	NA	4 m	1 points/30s

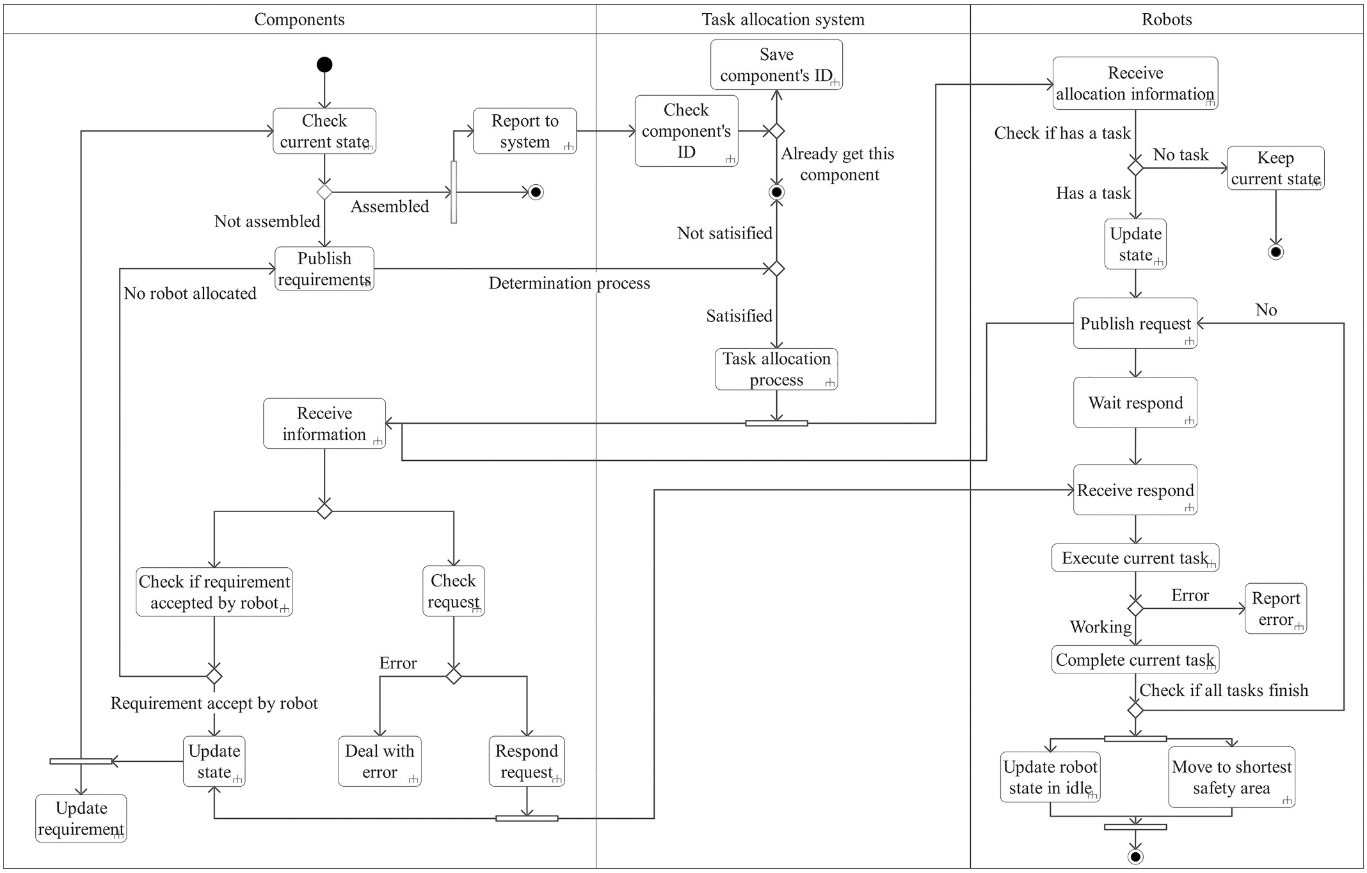


Fig. 9. Simulation process as deployed by our system.

and the next floor can be constructed only when the current floor is completed. The corresponding python-code for this simulation process is given in Listing 1.

Listing 1. Python-code that shows how the system determines when to move to the next level of construction.

```
def CheckFloor(currentFloorComponents, currentTotalNodesNumbers):
    #Iterate through all components of the current level
    for component in currentFloorComponents:
        #check component's node
        for node in component[1]:
            #check if node belong to this level and component in
              station
            if node.nodeLevel == currentFloor and node.currentState
              == 'Assembled' and node not in finishedNode:
                #add node to finished list
                finishedNodeList.append(node)
        #if all nodes in this floor assembled
    if len(finishedNodeList) == currentTotalNodesNumbers:
        currentFloor += 1
        return currentFloor
```

When the construction simulation starts, in the first step (top-left of Fig. 9), SCO components on the site publish their requirements to the interactive platform, and the system determines whether the current requirements of the component can be met based on the determination process. If the request meets the requirements (middle of Fig. 9), the system determines which task type belongs to transport, positioning, and assembly, then adds the construction task to the array of the corresponding construction task, and records the number of times the task was published, as well as its priority. If the request does not meet the requirement, the request is not accepted in this round.

When a construction task is ranked according to priority, it is then issued to the robot for matching according to priority. Using the free market approach (Section 3.3), the robot will match whether or not it can complete the build task based on its performance, and if so, calculate the time required to complete the task and provide feedback to the interactive platform. The interactive platform selects the robot with the shortest predicted construction time based on the robot's bidding price, and then assigns the component to the robot, establishing the relationship between the robot and the component. When all tasks are assigned, the state of the component changes to the next state, the robot state changes from idle to working, and the robots will each start executing the task. When the task is completed, the component state changes to the next state, and if the component is incomplete, a new request is sent to the interactive platform. If no new component is received, the robot moves to the yard nearest to the component to secure the construction area. While updating the status and requirements, the component and

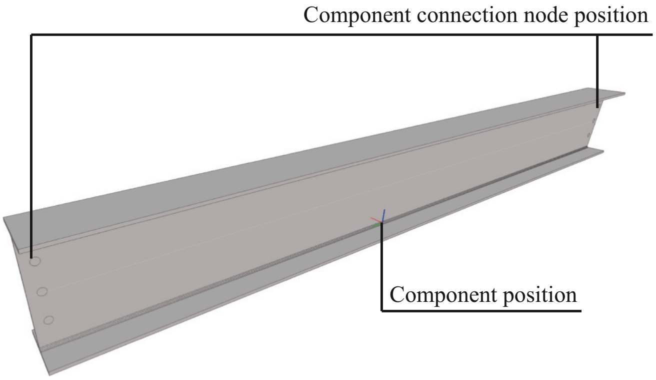


Fig. 10. Definition of the available positions for a component.

the robot need to be updated with the latest position information.

In each task step, the platform prints out the floor where this construction step is situated, the total number of steps performed, the tasks to be performed, the components being performed, the robots performing the task, and the time needed to complete the task. Listing 2 shows an example of how this information is printed.

Listing 2. The system displays the status of the construction process simulation.

```
-----------------------------------------------
Start a new construction step, NO.20, in 2 floor
-----------------------------------------------
Get components requirement from site

Publish transport requirements to robot
TransportRobot12 is working for component:3$PVtuH856a05znieudXmh,
  total work time is 15.04s
Current progress: Robot is unloading
Component: 3$PVtuH856a05znieudXmh is arrived, TransportRobot12 is
  free for next work
TransportRobot11 is working for component:0cj3XrufL6dAjrVT48UNpN,
  total work time is 30.69s
Current progress: Robot is unloading
Component: 0cj3XrufL6dAjrVT48UNpN is arrived, TransportRobot11 is
  free for next work

Publish positioning requirements to robot
PositioningRobot22 is positioning for nodes:2
  G$VGn2rT60B9v8wrT2c$P1Z4TeylFjEWxSPs$0FQ5I7,2
  tuDcUp6P0i992YTExwrg11Z4TeylFjEWxSPs$0FQ5I7, total work time is
  66.01s
Current progress: PositioningRobot22 is coming

Publish assembly requirements to robot
AssemblyRobot32 is finding nodes: 2
  G$VGn2rT60B9v8wrT2c$P1Z4TeylFjEWxSPs$0FQ5I7 and 2
  tuDcUp6P0i992YTExwrg11Z4TeylFjEWxSPs$0FQ5I7, total work time is
  66.11s
Current progress: AssemblyRobot32 is coming
```

At the end of each task step, the construction status of the floor is judged, and if the construction of that floor is judged to be complete, the next floor will start simulation. When the current floor is the last floor and the construction of that floor is finished, the number of completed components is verified, and the construction is complete if all components are connected.

5.4. Summary: trial simulation results

For the construction simulation process and system, two scenarios were designed to compare the impact on the construction schedule with a different number and types of robots participating. We used the virtual robots in Table 1 for simulation. In the first scenario, only robot 1 for each type of construction task is included, and the performance of those robots is assessed. In the second scenario, each type of construction task can be performed by two robots (both robot 1 and robot 2), and each robot can perform all construction tasks, but the performance of the robot is varied. In Table 2, we compare the construction step, construction time, and construction time per robot spent in these two scenarios to see the coordination of multiple robots in construction.

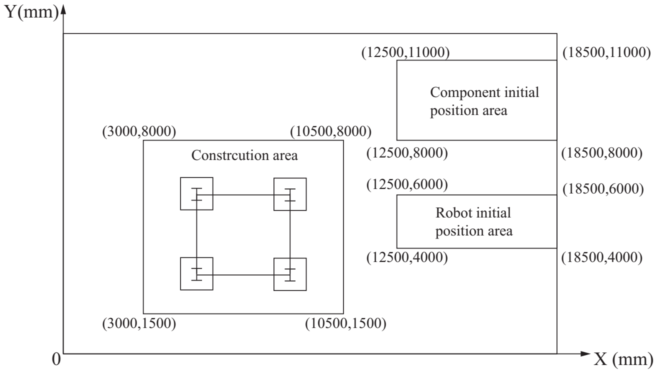


Fig. 11. Initial position plane showing all positions of all available components and robots.

Table 2
Comparison of scenario 1 and scenario 2 in terms of total steps and total time.

		Scenario 1	Scenario 2
Transport robot 1	Total steps	26	12
	Total time	200 s	52 s
Transport robot 2	Total steps	Not involved	12
	Total time	Not involved	102 s
Positioning robot 1	Total steps	40	20
	Total time	2543 s	1270s
Positioning robot 2	Total steps	Not involved	20
	Total time	Not involved	668 s
Assembly robot 1	Total steps	42	20
	Total time	7343 s	3700 s
Assembly robot 2	Total steps	Not involved	20
	Total time	Not involved	1850s
Overview of construction	Total steps	49	27
	Total time	9868 s	5072 s

We can find that when two robots work in collaboration, their construction steps and construction time are reduced and the construction efficiency is improved. This simple sample is mainly used to verify that the method of this paper achieves the coordination of the robots in the construction process.

6. Conclusion and future works

In this paper, we relied on the concept of SCOs, a new definition of construction components, to realize a method that can transform construction tasks into tasks that can be allocated to robots. This article furthermore aimed to enable and achieve robot coordination. After a literature review on the topic, we have defined a method, which was afterwards implemented and tested on a mock-up use case.

In our method, we rely heavily on SCOs. We give them construction state and connection properties in our simulation system, while allowing the components to make construction demand release and state transfers. We implemented the method in a simple Python-based simulation environment that is able to load IFC data from widely available BIM software.

The SCO-based approach, which assigns awareness, communicativeness and autonomy properties to components, enables a prefabricated construction method based on component requirements. This makes it possible to manage the construction process at a smaller scale, making it more accurate and controllable. At the same time, smart components can achieve autonomous interaction with other objects (e.g. robots, other components) or systems. Also, SCO component state can be automatically updated, providing a possible method for controlling robots to achieve automatic construction.

IFC4 can provide accurate and comprehensive information, realizing a method of unified description and delivery of BIM information. Through IFC, we not only obtain the physical information of the building components, but also the connection relationships between the components and the construction method. This is useful data that can easily be collected from available BIM models and used for robot task allocation planning.

Future research will be based on the approach proposed in this paper to do the following work: 1. implementation of the method in BIM models with more complex component types and connection methods; 2. more detailed descriptions and definitions of specific construction methods and robots, making it possible to use them for realistic robot model simulations; 3. experiment with various algorithms to optimize the construction simulation process based on this method, and 4. visualization of the construction simulation process in a user-friendly interface.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The support provided by China Scholarship Council (No.202007720036) during a study of Zhu Aiyu to Eindhoven University of Technology is acknowledged.

References

[1] B. García de Soto, I. Agustí-Juan, J. Hunhevicz, S. Joss, K. Graser, G. Habert, B. T. Adey, Productivity of digital fabrication in construction: cost and time analysis of a robotically built wall, Autom. Constr. 92 (2018) 297–311, https://doi.org/10.1016/j.autcon.2018.04.004.

[2] D. Castro-Lacouture, Construction automation, in: Springer Handbook of Automation, Springer, 2009, pp. 1063–1078, https://doi.org/10.1007/978-3-540-78831-7.

[3] J. Buchli, M. Giftthaler, N. Kumar, M. Lussi, T. Sandy, K. Dörfler, N. Hack, Digital in situ fabrication - challenges and opportunities for robotic in situ fabrication in architecture, construction, and beyond, Cem. Concr. Res. 112 (2018) 66–75, https://doi.org/10.1016/j.cemconres.2018.05.013, sI: Digital concrete 2018.

[4] J.M. Davila Delgado, L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, H. Owolabi, Robotics and automated systems in construction: understanding industry-specific challenges for adoption, J. Build. Eng. 26 (2019) 100868, https://doi.org/10.1016/j.jobe.2019.100868.

[5] N. Melenbrink, J. Werfel, A. Menges, On-site autonomous construction robots: towards unsupervised building, Autom. Constr. 119 (2020) 103312, https://doi.org/10.1016/j.autcon.2020.103312.

[6] T. Bock, T. Linner, Complex Products in Other Industries and Relevance of Fixed-Site/On-Site Manufacturing Technology, Cambridge University Press, 2015, pp. 125–155, https://doi.org/10.1017/CBO9781139924146.006.

[7] F. Craveiro, J.P. Duarte, H. Bartolo, P.J. Bartolo, Additive manufacturing as an enabling technology for digital construction: a perspective on construction 4.0, Autom. Constr. 103 (2019) 251–267, https://doi.org/10.1016/j.autcon.2019.03.011.

[8] X. Li, G.Q. Shen, P. Wu, H. Fan, H. Wu, Y. Teng, RBL-PHP: simulation of lean construction and information technologies for prefabrication housing production, J. Manag. Eng. 34 (2018) 1–18, https://doi.org/10.1061/(ASCE)ME.1943-5479.0000577.

[9] Z. Feng, G. Hu, Y. Sun, J. Soon, An overview of collaborative robotic manipulation in multi-robot systems, Annu. Rev. Control. 49 (2020) 113–127, https://doi.org/10.1016/j.arcontrol.2020.02.002.

[10] R.A. Knepper, T. Layton, J. Romanishin, D. Rus, Ikeabot: An autonomous multi-robot coordinated furniture assembly system, in: 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 855–862, https://doi.org/10.1109/ICRA.2013.6630673.

[11] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, 2006, https://doi.org/10.1117/1.2819119.

[12] Z. Yan, N. Jouandeau, A.A. Cherif, A survey and analysis of multi-robot coordination, Int. J. Adv. Robot. Syst. 10 (2013) 399, https://doi.org/10.5772/57313.

[13] J. Chen, Y. Yang, L. Wei, Research on the approach of task decomposition in soccer robot system, in: 2010 International Conference on Digital Manufacturing Automation Vol. 2, 2010, pp. 284–289, https://doi.org/10.1109/ICDMA.2010.33.

[14] O. Hamdi, Advanced Work Packaging from Project Definition Through Site Execution: Driving Successful Implementation of WorkFace Planning, Ph.D. thesis, University of Texas, 2013. URL: http://hdl.handle.net/2152/21384.

[15] W. Jeong, S. Chang, J. Son, J.-S. Yi, BIM-integrated construction operation simulation for just-in-time production management, Sustainability 8 (2016) 1–25, https://doi.org/10.3390/su8111106.

[16] B.P. Gerkey, M.J. Matarić, A formal analysis and taxonomy of task allocation in multi-robot systems, Int. J. Robot. Res. 23 (2004) 939–954, https://doi.org/10.1177/0278364904045564.

[17] W. Lu, T. Olofsson, Building information modeling and discrete event simulation: towards an integrated framework, Autom. Constr. 44 (2014) 73–83, https://doi.org/10.1016/j.autcon.2014.04.001.

[18] Y. Niu, W. Lu, K. Chen, G.G. Huang, C. Anumba, Smart construction objects, J. Comput. Civ. Eng. 30 (2016), 04015070, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000550.

[19] X. Li, G.Q. Shen, P. Wu, F. Xue, H. Lin Chi, C.Z. Li, Developing a conceptual framework of smart work packaging for constraints management in prefabrication housing production, Adv. Eng. Inform. 42 (2019) 100938, https://doi.org/10.1016/j.aei.2019.100938.

[20] Smith, The contract net protocol: high-level communication and control in a distributed problem solver, in: IEEE Transactions on Computers C-29, 1980, pp. 1104–1113, https://doi.org/10.1109/TC.1980.1675516.

[21] A.T. Stentz, M.B. Dias, A Free Market Architecture for Coordinating Multiple Robots, Technical Report CMU-RI-TR-99-42, Carnegie Mellon University, Pittsburgh, PA, 1999, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.68.9674.

[22] T. Bock, T. Linner, Transition and Technological Reorientation Towards Integrated On-Site Manufacturing Vol. 3, Cambridge University Press, 2016, pp. 291–301, https://doi.org/10.1017/CBO9781139872041.004.

[23] T. Bock, T. Linner, Single-Task Construction Robots by Category Vol. 3, Cambridge University Press, 2016, pp. 14–290, https://doi.org/10.1017/CBO9781139872041.002.

[24] T. Bock, T. Linner, Integrated Automated/Robotic On-site Factories, Cambridge University Press, 2016, pp. 1–7, https://doi.org/10.1017/CBO9781139872027.002.

[25] T. Bock, The future of construction automation: technological disruption and the upcoming ubiquity of robotics, Autom. Constr. 59 (2015) 113–121, https://doi.org/10.1016/j.autcon.2015.07.022.

[26] F. Nigl, S. Li, J.E. Blum, H. Lipson, Structure-reconfiguring robots: autonomous truss reconfiguration and manipulation, IEEE Robot. Autom. Mag. 20 (2013) 60–71, https://doi.org/10.1109/MRA.2012.2201579.

[27] Y. Yoon, D. Rus, Shady3d: A robot that climbs 3d trusses, in: Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 4071–4076, https://doi.org/10.1109/ROBOT.2007.364104.

[28] J.J. Kuffner, S.M. LaValle, Rrt-connect: An efficient approach to single-query path planning, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065) Vol. 2, 2000, pp. 995–1001, https://doi.org/10.1109/ROBOT.2000.844730.

[29] A. Gandia, S. Parascho, R. Rust, G. Casas, F. Gramazio, M. Kohler, Towards automatic path planning for robotically assembled spatial structures, in: Robotic Fabrication in Architecture, Art and Design 2018, Springer International Publishing, 2018, pp. 59–73, https://doi.org/10.1007/978-3-319-92294-2_5.

[30] K. Kotay, D. Rus, The inchworm robot: a multi-functional system, Auton. Robot. 8 (2000) 53–69, https://doi.org/10.1023/A:1008940918825.

[31] Y. Terada, S. Murata, Automatic modular assembly system and its distributed control, Int. J. Robot. Res. 27 (2008) 445–462, https://doi.org/10.1177/0278364907085562.

[32] B. Jenett, A. Abdel-Rahman, K. Cheung, N. Gershenfeld, Material-robot system for assembly of discrete cellular structures, IEEE Robot. Autom. Lett. (2019) 1, https://doi.org/10.1109/LRA.2019.2930486.

[33] J. Werfel, K. Petersen, R. Nagpal, Designing collective behavior in a termite-inspired robot construction team, Science 343 (2014) 754–758, https://doi.org/10.1126/science.1245842.

[34] D. Wood, M. Yablonina, M. Aflalo, J. Chen, B. Tahanzadeh, A. Menges, Cyber physical macro material as a UAV [re] configurable architectural system, in: Robotic Fabrication in Architecture, Art and Design 2018, Springer International Publishing, 2018, pp. 320–335, https://doi.org/10.1007/978-3-319-92294-2_25.

[35] M. Taghavi, T. Kinoshita, T. Bock, Design, modelling and simulation of novel hexapod-shaped passive damping system for coupling cable robot and end effector in curtain wall module installation application, in: M. Al-Hussein (Ed.), Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC), International Association for Automation and Robotics in Construction (IAARC), Banff, AB, Canada, 2019, pp. 665–671, https://doi.org/10.22260/ISARC2019/0089.

[36] B. Felbrich, N.F. h M. Prado, S. Saffarian, J. Solly, L. Vasey, J. Knippers, A. Menges, Multi-Machine Fabrication: An Integrative Design Process Utilising an Autonomous UAV and Industrial Robots for the Fabrication of Long Span Composite Structures, Zenodo, 2017, pp. 248–269, https://doi.org/10.5281/zenodo.2667782.

[37] Yang Yi, Pan Mi, Pan Wei, 'Co-evolution through interaction' of innovative building technologies: The case of modular integrated construction and robotics, Automation in Construction 107 (0926-5805) (2019), https://doi.org/10.1016/j.autcon.2019.102932.

[38] S. Isaac, M. Curreli, Y. Stoliar, Work packaging with bim, Autom. Constr. 83 (2017) 121–133, https://doi.org/10.1016/j.autcon.2017.08.030.

[39] H.-W. Wang, J.-R. Lin, J.-P. Zhang, Work package-based information modeling for resource-constrained scheduling of construction projects, Autom. Constr. 109 (2020) 102958, https://doi.org/10.1016/j.autcon.2019.102958.

[40] X. Li, G.Q. Shen, P. Wu, T. Yue, Integrating building information modeling and prefabrication housing production, Autom. Constr. 100 (2019) 46–60, https://doi.org/10.1016/j.autcon.2018.12.024.

[41] Y. Niu, W. Lu, D. Liu, K. Chen, C. Anumba, G.G. Huang, An sco-enabled logistics and supply chain–management system in construction, J. Constr. Eng. Manag. 143 (2017), 04016103, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001232.

[42] Y. Niu, W. Lu, D. Liu, K. Chen, F. Xue, A smart construction object (sco)-enabled proactive data management system for construction equipment management, in: Computing in Civil Engineering 2017, 2017, pp. 130–138, https://doi.org/10.1061/9780784480830.017.

[43] X. Li, C. Wu, P. Wu, L. Xiang, G.Q. Shen, S. Vick, C.Z. Li, Swp-enabled constraints modeling for on-site assembly process of prefabrication housing production, J. Clean. Prod. 239 (2019) 117991, https://doi.org/10.1016/j.jclepro.2019.117991.

[44] R. Zlot, A. Stentz, Market-based multirobot coordination for complex tasks, Int. J. Robot. Res. 25 (2006) 73–101, https://doi.org/10.1177/0278364906061160.

[45] B. Arbanas, A. Ivanovic, M. Car, T. Haus, M. Orsag, T. Petrovic, S. Bogdan, Aerial-ground robotic system for autonomous delivery tasks, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 5463–5468, https://doi.org/10.1109/ICRA.2016.7487759.

[46] B.P. Gerkey, M.J. Mataric, Sold!: auction methods for multirobot coordination, IEEE Trans. Robot. Autom. 18 (2002) 758–768, https://doi.org/10.1109/TRA.2002.803462.

[47] M. Bernardine Dias, R. Zlot, N. Kalra, A. Stentz, Market-based multirobot coordination: a survey and analysis, Proc. IEEE 94 (2006) 1257–1270, https://doi.org/10.1109/JPROC.2006.876939.

[48] E. Tuci, M.H. Alkilabi, O. Akanyeti, Cooperative object transport in multi-robot systems: A review of the state-of-the-art, in: Frontiers Robotics AI 5, 2018, https://doi.org/10.3389/frobt.2018.00059.

[49] L.E. Parker, D. Rus, G.S. Sukhatme, Multiple mobile robot systems, in: Springer Handbook of Robotics, Springer International Publishing, 2016, pp. 1335–1384, https://doi.org/10.1007/978-3-319-32552-1_53.

[50] C.Z. Li, R.Y. Zhong, F. Xue, G. Xu, K. Chen, G.G. Huang, G.Q. Shen, Integrating rfid and bim technologies for mitigating risks and improving schedule performance of prefabricated house construction, J. Clean. Prod. 165 (2017) 1048–1062, https://doi.org/10.1016/j.jclepro.2017.07.156.

[51] R. Bostelman, S. Foufou, S. Legowik, T. Hong, Mobile manipulator performance measurement towards manufacturing assembly tasks, in: Proceedings of the 13th IFIP International Conference on Product Lifecycle Management (PLM16), Columbia, SC, 2016, pp. 411–420, https://doi.org/10.1007/978-3-319-54660-5_37.

[52] M. Hattig, I. Horswill, J. Butler, Roadmap for mobile robot specifications, in: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453) Vol. 3, 2003, pp. 2410–2414, https://doi.org/10.1109/IROS.2003.1249231.