# Cocoon | Collaborative Construction

***A multi-robot system for complex tasks on construction sites: Reference case of a brick wall construction***

Doğa Su Kıralioğlu
RWTH Aachen University
Construction Robotics Master Program
Prototyping Project WS 2023-24
Chair Design Computation

February 29, 2024

## Disclaimer

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent porttitor arcu luctus, imperdiet urna iaculis, mattis eros. Pellentesque iaculis odio vel nisl ullamcorper, nec faucibus ipsum molestie. Sed dictum nisl non aliquet porttitor. Etiam vulputate arcu dignissim, finibus sem et, viverra nisl. Aenean luctus congue massa, ut laoreet metus ornare in. Nunc fermentum nisi imperdiet lectus tincidunt vestibulum at ac elit.

## Copyright



2024
dogasukiralioglu@gmail.com

Construction Robotics Master’s Program | Winter Semester 2023-24. RWTH Aachen University. Special thanks to Prof. Jakob Beetz.

## Contact

Kastanianweg 35
Kawo 2, 1032
Contact: dogasukiralioglu@gmail.com

# Table of Contents

# 1 Introduction

Throughout the last decades, the yearning for originality has only been increasing. Many artists believed that the way to have a breakthrough and achieve originality in art was only possible through a new way of making art. A new architecture can be made possible by a new way of building, a new way of making architecture. The introduction of robotics into the domain of construction holds the promise of this possibility.

The benefits of the introduction of robotic systems into the architectural domain is nevertheless not limited to this promise. Robotic systems on construction sites open up new possibilities for both artistic and technical advancements. This revolution would not only increase efficiency in a sector that is in dire need of it, but also improve working conditions for workers, and even allow for construction projects in extreme environments that would previously be impossible.

make these two paragraphs into one

Still, the implementation of on-site robotic automation in the construction industry so far remains very limited. As stated by Melenbrink et al. (2020), increasing robotic automation on construction sites could in return reduce injury rates, increase efficiency in repetitive tasks, and help enable construction in environments in which construction is currently impossible or too dangerous, such as disaster relief or exoplanet construction.

Another reason for construction industry being the perfect ground for robotic advancements is the 4D principle. This principle suggests that robotic systems are especially fit for implementation for jobs or environments that are dirty, dull, dangerous or difficult. Many of the tasks on a construction site correspond to at least one of these criteria, which is why economists and investors soon expect a robotic revolution in the Architecture, Engineering and Construction (AEC) industry [2].

While these immediate practical advantages can be observed already in the industry, the implementation of robotic systems in construction holds promise on a much larger scale. With the recent advancements in technology, automation is getting more and more widespread in all of the industries. The most efficient way of producing anything is standardization and mass production. Still, for certain practices that are ancient and painfully human, such as cooking and architecture, complete standardization and mass production is not an option that is acceptable.

These practices are very tightly woven together with culture, tradition, and community so that they have a bi-directional affect on each other. This reality, together with the idea that the changes in the way of making construction will have a direct affect on the results of the process, which is the architecture that is made, makes it abundantly clear that how we handle the introduction of robots into the architectural domain will determine how our architecture will be in the upcoming decades.

# 2 Literature Review

## 2.1 State of the Art

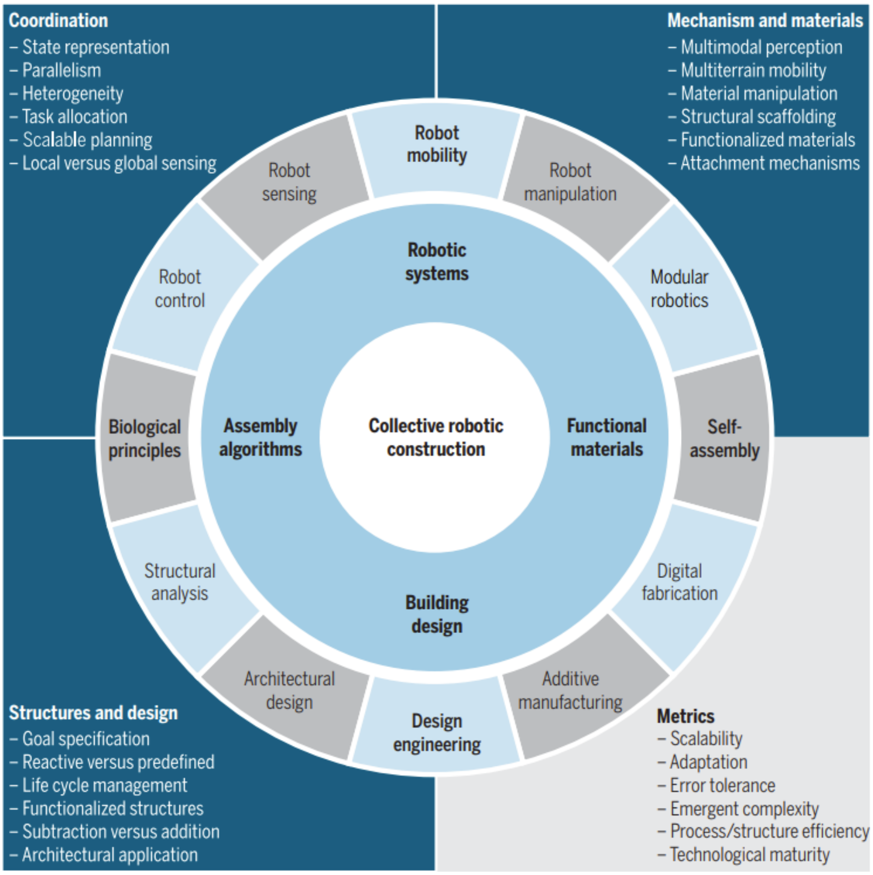


**Figure 1: Diagram showing how collective robotic construction depends on and interacts with many other fields.**

The implementation of robotic systems in construction sectors have already proven itself fruitful in several case studies. Some of the examples of these advantages in implementation include, SAM100 the robotic bricklaying aid, which assists workers by helping to carry the load of assembly blocks. While this robotic implementation is on the lower end in terms of level of autonomy, it's practical safety mechanisms make for a good example solution for situations where workers have to carry heavy loads. (4)

Another example that makes the construction workers life easier is HILTI Jaibot (5). This drilling robot takes location information from IFC models which include specially designed IFC objects, and executes the drilling operation according to the retrieved location information using its sensors. This implementation case is not only admirable for providing an automated solution to a very common construction task, but also for relieving the construction workers from this highly strenuous activity.

To go one step further, the example of [6] demonstrates the construction possibilities with autonomous robots in environments unsuitable for human workers. In exoplanatery construction, lunar in this case, the robotic systems are not an option but usually a necessity. Still, in construction and site preparation tasks as demonstrated in (6), there are multiple factors that determine the suitability of a multi robot systems that has to be considered on top of traditional metrics, such as the lightweightness or multi-functionality of robots.

## 2.2 Research Gap

Introduction of robotic systems into construction sites has been taking up speed and receiving more and more attention throughout the last decade. Still, there are many gaps to be filled for a complete autonomous workflow to be possible on the construction site. On of the major gaps, as stated by (1) is the lack of solutions to coordinate multiple robots with different capabilities that operate on the vast amounts of different construction tasks. Filling this gap is crucial for a multi-robot system to be functional on any construction site. The shift in the understanding of construction site planning and scheduling from temporal planning to spatio-temporal planning, such as LEAN scheduling, provide a good starting ground for robotic systems scheduling solutions on construction sites.

Another point for major improvement lies in the myriad amounts of data that is being produced for any and every construction project. Data coming from IFC/ BIM models are not utilized anywhere close to their full potential. Even though there are some use cases with limited implementation, such as Hilti Jaibot which utilizes IFC data for locations (5), there is still a major potential for IFC data utilization for fabrication data and robot motion generation.

One other gap that needs to be taken into consideration before a fully automated MRS on the construction site is possible, is lack of reference processes for certain common construction tasks. Tasks such as drilling, brick laying, painting, tiling and many more, are fundemnetal to many construction projects all over the world. Producing reference processes for these tasks would open up vast possibilities for immediate implementation of MRS on construction sites. Since subcontracting is already a common practice within the industry, pinpointed solutions automating or assisting workers for very specific processes, such as Hilti Jaibot and SAM100 respectively, are embraced in the industry and adapted relatively faster. These task specific solutions would also be a crucial chain for a complete fully autonomous MRS on construction sites.

# 3 Prototype

## 3.1 Complete Workflow of IFC to Fabrication

This prototype assumes the existence of a standard IFC Model that was prepared for the hypothetical single storey residential project. One of the aims of this prototype is to demonstrate the possibilities of information retrieval from a standard IFC model, and extents to which this information can be utilized for several application areas such as fabrication, planning, and scheduling.

Fabrication data generation including the robot motions, does not require any additional input other than this IFC Model. The whole workflow can be divided into two parts: IFC information retrieval with Python, and Fabrication Data Generation

## 3.2 IFC Information Retrieval with Python

The information stored within IFC model is retrieved in this step using Python and IFC OpenShell API. This information is then translated into fabrication data and robot motions in the upcoming steps.

```

# Import necessary libraries and functions.

import ifcopenshell
import ifcopenshell.api
from ifcopenshell.api import run
import ifcopenshell.util.sequence
import ifcopenshell.util.placement
from collections import Counter

### DEFINITIONS ###

# Retrieve all IFC Tasks and their hierarchical information.

def print_all_tasks(ifc_model):
    printed_tasks = set()
    leaf_tasks = {}

    def print_task_with_gap(task, level):
        print("␣␣␣" * level + "Task␣ID:", task.id())
        print("␣␣␣" * level + "Task␣Name:", task.Name if hasattr(task
            , "Name") else "N/A")
        print("␣␣␣" * level + "-" * 40)
        printed_tasks.add(task.id())

        if level in leaf_tasks:
            leaf_tasks[level].append(task)
        else:
            leaf_tasks[level] = [task]


    def print_nested_tasks(tasks, current_level=0):
        for task in tasks:
            if task.id() not in printed_tasks:
                print_task_with_gap(task, current_level)
                nested_tasks = ifcopenshell.util.sequence.
                    get_nested_tasks(task)
```

```
                if nested_tasks:
                    print_nested_tasks(nested_tasks, current_level + 1)

    print_nested_tasks(tasks)
    return leaf_tasks

# Find the highest level in hirearchy, since the tasks on this level will be the first tasks to be completed in the schedule.

def print_task_levels(leaf_tasks):
    for task in leaf_tasks[max_level]:
        print(f"Task␣ID:␣{task.id()},␣Task␣Name:␣{task.Name␣if␣hasattr(task,␣'Name')␣else␣'N/A'},␣Level:␣{max_level}")

# Find all relevant tasks for the desired IFC product, using keywords such as "wall, slab, beam" etc.

def filter_tasks_by_keyword(tasks, keyword):
    filtered_tasks = []
    for task_list in tasks.values():
        for task in task_list:
            if keyword.lower() in task.Name.lower():
                filtered_tasks.append(task)
    print('␣␣␣␣' f'{keyword}␣tasks')
    for task in filtered_tasks:
        print(f"Task␣ID:␣{task.id()},␣Task␣Name:␣{task.Name␣if␣hasattr(task,␣'Name')␣else␣'N/A'},␣Level:␣{max_level}")
    return filtered_tasks

# Get tasks predecessor information.

def get_predecessor(task):
    # Use the BlenderBIM API utility functions
    predec = ifcopenshell.util.sequence.get_sequence_assignment(task, sequence='predecessor')
    return predec

# Find tasks without predecessors, since these tasks have to be carried out first. This definition helps avoid scheduling problems, such as scheduling a wall finishing job before the wall assembly job.

def find_initial_tasks(tasks):
    tasks_wo_predecs = []
    for task in tasks:
        predecs = get_predecessor(task)
        if not predecs:
            tasks_wo_predecs.append(task)
    return tasks_wo_predecs

# Find tasks' products, and write them into a new IFC file

def find_task_products(tasks):
    outputs = []
    for task in tasks:
        products = ifcopenshell.util.sequence.get_direct_task_outputs(task)
        for product in products:
            walls_model.add(product)
            outputs.append(product)
    return outputs

# Find center points of products. This location will be the starting point for each robot for their respective tasks.

def find_center_point(product):
    matrix = ifcopenshell.util.placement.get_local_placement(product.ObjectPlacement)
    # get location
```

```
    location = matrix[:3, 3]
    # Calculate center point
    center_point = tuple(location)
    return center_point

# Create a dictionary with tasks and center points of their respective products.

def match_tasks_w_points(tasks):
    center_points = {}
    products = find_task_products(tasks)
    for product in products:
        center_point = find_center_point(product)
        center_points[product] = center_point
    return center_points

# Define a main function to create the robot world and a dictionary which stores the locations of tasks.

if __name__=="__main__":
    directory = "/home/doga/src/mrta/"
    ifc_file_path = os.path.join(directory, "basemodel.ifc")

    # Open the IFC model, and retrieve necessary information using respective definitions.

    model = ifcopenshell.open("basemodel.ifc")
    tasks = model.by_type("IfcTask")
    leaf_tasks = print_all_tasks(model)
    max_level = max(leaf_tasks.keys())
    wall_tasks = filter_tasks_by_keyword(leaf_tasks, "wall")
    initial_wall_tasks = find_initial_tasks(wall_tasks)

    ### create robot world ###

    # Create a new IFC model containing only the relevant IFC products and their 3D geometric representation.

    walls_model = ifcopenshell.file(schema=model.schema)
    walls_project = run("root.create_entity", walls_model, ifc_class="IfcProject", name="Walls")

    # TIP: 3D geometries will not be represented in the file if you don’t assign units.

    length_unit = "MILLIMETER"
    run("unit.assign_unit", walls_model, length_unit)

    context = run("context.add_context", walls_model, context_type="Model")
    body = run("context.add_context", walls_model, context_type="Model",
        context_identifier="Body", target_view="MODEL_VIEW", parent=context)

    # Add relevant products into the model using find_task_products definition.

    walls = find_task_products(initial_wall_tasks)

    # Write the new IFC model.

    walls_model.write(’walls_model.ifc’)

    ### create task-location dictionary ###

    match_tasks_w_points(initial_wall_tasks)
```

### 3.3 Fabrication Data Generation with Cocoon Discrete Assembly Plug-in for Rhino Grasshopper

# 4 Referencing Citations

Citations styles that make authors and publication year visible are preferable too numeric styles.

APA is an example of such a style.

This statement requires citation (Borrmann et al., 2021).

(Melenbrink et al., 2020)

This statement requires multiple citations (Schulz et al., 2023).

This short citation is in the margin[1].

This long citation is in the margin[2].

This statement has an in-text citation: Kremer (2022).

[1] Göbels (2022)

[2] Werbrouck, J., Schulz, O., Oraskari, J. T., Mannens, E., Pauwels, P., & Beetz, J. (2023). A generic framework for federated CDEs applied to Issue Management. *Advanced engineering informatics*, *58*, 102136. https://doi.org/10.1016/j.aei.2023.102136

# 5 Link Examples

This is a URL link: DuckDuckGo.

This is a email link: example@example.com.

This is a monospaced URL link: https://duckduckgo.com.

Links can be clicked in the PDF to navigate to the linked website or email address.

# 6 Equation

$$\cos^3\theta = \frac{1}{4}\cos\theta + \frac{3}{4}\cos 3\theta \tag{1}$$

This statement automatically references the equation above using its label: Equation 1.

# Reference List

Borrmann, A., König, M., Koch, C., & Beetz, J. (Eds.). (2021). *Building Information Modeling: Technologische Grundlagen und industrielle Praxis* (2. Auflage). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-33361-4

Göbels, A. (2022). Enabling object-based documentation of existing bridge inspection data using Linked Data. *Proceedings of 33. Forum Bauinformatik / Herausgeber: Slepicka, S.; Kolbeck, L.; Esser, S.; Forth, K.; Noichl, F.; Schlenger, J.*, 148–155. https://publications.rwth-aachen.de/record/854732

Kremer, N. C. (2022). Tracking model checks in information delivery controlling processes. *Proceedings of 33. Forum Bauinformatik 07. – 09. September 2022 / Herausgeber: Martin Slepicka, Lothar Kolbeck, Sebastian Esser, Kasimir Forth, Florian Noichl, Jonas Schlenger*, 45–52. https://publications.rwth-aachen.de/record/862313

Melenbrink, N., Werfel, J., & Menges, A. (2020). On-site autonomous construction robots: Towards unsupervised building. *Automation in construction*, *119*, 103312.

Schulz, O., Oraskari, J. T., & Beetz, J. (2023). Lessons Learned from Designing and Using bcfOWL. *[11th Linked Data in Architecture and Construction Workshop, LDAC2023, 2023-06-15 - 2023-06-16, Matera, Italy]*, 12 Seiten. https://doi.org/10.18154/RWTH-2023-05985

Werbrouck, J., Schulz, O., Oraskari, J. T., Mannens, E., Pauwels, P., & Beetz, J. (2023). A generic framework for federated CDEs applied to Issue Management. *Advanced engineering informatics*, *58*, 102136. https://doi.org/10.1016/j.aei.2023.102136

# 7 Compilation of initial references

- possible robots:

  Menzi Muck M545 (used in HEAP by ETH)

  Neobotix MMO-500 (used in Development of BIM-integrated construction robot task planning and simulation system)

  Brokk (used in many studies)

  MULE135 (produced by Construction Robotics USA): has 0 autonomy, could be used w other agents

  SAM100 (produced by Construction Robotics USA) (Semi-Automated Mason)

- Kim, S., Peavy, M., Huang, P. C., Kim, K. (2021). Development of BIM-integrated construction robot task planning and simulation system. Automation in Construction, 127, 103720.

  This study extends BIM to incorporate robot task planning and generate detailed motions conducting construction tasks.

  Unlike ordinary construction scheduling, 4D BIM explicitly visualizes the construction plan by linking spatial and temporal aspects of a project.

  However, insufficient efforts have been made in extending the boundary of BIM to construction robot task planning. Beyond simple visualization of robot objects, a new BIM tool for construction robotization should support the generation of elemental motions (e.g., grasping, moving, turning, picking) of robots considering contextual information presented in BIM. The performances (e.g., duration, safety, cost) associated with executing the robot task plan should be presented.

Even with the wide adoption of BIM technologies, it is unlikely that architects or construction planners could manually create Level 3 Elemental Motions of robots due to the myriad amount of manual efforts required. Even with the notable efforts in construction robotics, none of the recent studies [14–16,35] addressed the gap between construction and robotics by enabling construction robot task planning.

None of the previous studies properly addressed the fundamental challenge of generating elemental robot motion.

Therefore, this study imported spatiotemporal information from the BIM to generate a "world" for robot simulation. Similarly, non-building elements (e.g., workers, equipment, etc.) in a construction site are incorporated in the "world" where robot behaviors are created. Placements of these non-building elements should be considered when generating movements and motions of robots. Lastly, the system should be able to generate detailed elemental motions of robots based on the input, and the expected outcomes of the robot behaviors should be reported back to the user.

A building model created in Autodesk Revit is first exported as an IFC file with IFC2x3 (AAA would be IFC 4.0 for me) Coordination View setting. The IFC-SDF converter then generates an SDF building model that can be inserted into a Gazebo world model. In addition to the building SDF model generated by the IFC-SDF converter, a mobile robot and non-building objects can be inserted in the world as Universal Robot Description Format (URDF) and SDF format, respectively.

The ROS framework provides an environment where a developer can combine numerous sub-processes called ROS nodes into an application package.

More diverse types of building components with complex shapes should be incorporated in the conversion process so that complicated building models and dynamic construction environments can be seamlessly imported into robot simulation environment.

- A Comprehensive Taxonomy for Multi-Robot Task Allocation G. Ayorkor Korsah, M. Bernardine Dias, Anthony Stentz

  Complex Task: A complex task is a multiply decomposable task for which there exists at least one decomposition that is a set of multi[agent]-allocatable subtasks. Each subtask in a complex task's decomposition may be simple, compound, or complex.

  capability constraints may define which robots are capable of performing which tasks. Capacity constraints can define how many tasks a given robot can perform at a time. Simultaneity constraints can specify that two tasks must be performed at the same time, while non-overlapping constraints may specify that they must not be performed at the same time, and precedence constraints may specify that one task must be performed before another.

Complex Dependencies (CD): These are task allocation problems for which the agent-task utilities have inter-schedule dependencies for complex tasks (in addition to any in-schedule and cross-schedule dependencies for simple or compound tasks). That is, the effective utility of an agent for a task depends on the schedules of other agents in the system in a manner that is determined by the particular task decomposition that is ultimately chosen. Thus, the optimal task decomposition cannot be decided prior to task allocation, but must be determined concurrently with task allocation. Furthermore, constraints may exist between the schedules of different agents.

- DONE On-site autonomous construction robots: Towards unsupervised building Nathan Melenbrinka,b,*, Justin Werfelb, Achim Menges

  This review presents a broad range of advancements in construction automation research, and finds that achieving fully autonomous construction in unstructured environments will require considerably more development in all three groups of construction tasks, as well as a particular emphasis on coordinating myriad construction tasks between different task-specific robots

  In the construction industry, the role of on-site robotic automation so far remains very limited. Increasing robotic automation on construction sites could carry substantial advantages such as reducing injury rates, handling repetitive tasks, and helping to enable construction in settings not currently feasible, e.g., for use in disaster relief, exoplanet construction, or other dangerous or challenging environments. A fully autonomous construction system, able to operate without supervision or intervention, would be best suited for these kinds of scenarios. Such a system would need to be able to handle unpredictable and changing conditions during the course of a project. Despite being heavily reliant upon mechanical operations, construction ranks as one of the least digitized industries [1]. Many tasks associated with the construction industry fulfill the canonical “dirty, dangerous and dull” criteria of tasks ripe for automation, which leads economists and investors to expect an imminent robotics revolution in the Architecture, Engineering and Construction (AEC) industry [2].

  While there are different systems for characterizing a robot’s level of autonomy (LoA), the most widely recognized metric is the scale from 0 (fully non-autonomous) to 5 (fully autonomous) that has been employed as a standard by the Society of Automotive Engineers (SAE) [17]. Work reviewed in this paper is characterized according to its level of autonomy in Section 5.

  Thomas Bock describes the current state of construction automation as consisting of two distinct paradigms, each represented by its own S-curve [13].

- Building Smart, Industry Foundation Classes (IFC) – An Introduction. https://tech nical.buildingsmart.org/standards/ifc/, 2020.

  IFC is a standard digital description of building and infrastructure models. As a vendor-neutral format, IFC files can be used to exchange project information between different platforms and processes without loss or misinterpretation of information.

- Open Source Robotics Foundation, SDFormat. http://sdformat.org/spec, 2020 (Accessed October 19, 2023).

  An SDF is an XML format describing objects and environments for robotics simulation and control

- Geometry Gym Pty Ltd, GeometryGymIFC. https://github.com/GeometryGym/Ge ometryGymIFC, 2021 (Accessed October 18, 2023).

  For this study, the converter was developed based on the open-source IFC parser GeometryGymIFC [43] to interface BIM and Gazebo which is a widely used physics-based robot simulation platform.

- Gazebo, Robot Simulation made Easy. http://gazebosim.org/, 2020 (Accessed October 19, 2020).

  widely used physics-based robot simulation platform.

- A. Ibrahim, A. Sabet, M. Golparvar-Fard, BIM-driven mission planning and navigation for automatic indoor construction progress detection using robotic ground platform, in: Proc. 2019 Eur. Conf. Comput. Constr., 2019, https://doi.org/ 10.35490/ec3.2019.195.

  presented an approach for mapping the as-built state of an indoor construction site. A robot's survey waypoints were first created in BIM and translated into coordinates in a robot navigation map. Then, the survey robot conducted LiDAR-based as-built modeling while avoiding obstacles in the moving paths.

- B.R.K. Mantha, M.K. Jung, B.G. de Soto, C.C. Menassa, V.R. Kamat, Generalized task allocation and route planning for robots with multiple depots in indoor building environments, Autom. Constr. 119 (2020) 103359.

  proposed a method to optimize task allocation and route-planning for multiple indoor building service robots.

- T. Bock, T. Linner, Robot Oriented Design, Cambridge University Press, 2015. [6] J.G. Everett, A.H. Slocum, Automation and robotics opportunities: construction versus manufacturing, J. Constr. Eng. Manag. (1994), https://doi.org/10.1061/ (ASCE)0733-9364(1994)120:2(443).

  Level Description Examples 1 Project Petrochemical plant, office building 2 Division Concrete, masonry, mechanical 3 Activity Drywall partition, concrete wall 4 Basic Task Connect, cut, measure, position 5 Elemental Motion Reach, grasp, travel 6 Orthopedics Muscle, bone, joint

- P.M. Teicholz, Labor-Productivity Declines in the Construction Industry: Causes and Remedies (a Second Look), AECbytes Viewp, 2013.

  Even with the adoption of three dimensional (3D) modeling and sensing technologies, the construction industry's labor productivity has declined 0.322012 while other non-farm industries' productivity grew 3.06

- N. Walker, Y. Jiang, M. Cakmak, P. Stone, Desiderata for Planning Systems in General-purpose Service Robots, ArXiv Prepr. ArXiv1907.02300, 2019.

  A general-purpose robot is equipped with multiple skills (e.g., scanning, navigation, arm movement), and its detailed motions implementing the skills and sequences between the skills can be flexibly programmed considering the characteristics of tasks and work environments

- Meschini, S., Iturralde, K., Linner, T., Bock, T. (2016). Novel applications offered by integration of robotic tools in BIM-based design workflow for automation in construction processes. In Proceedings of the CIB* IAARC W119 CIC 2016 Workshop.

  To address this deficiency, there is a need to integrate construction planning and robot task planning platforms. More specifically, they discussed the need and potential benefits that can be obtained by developing a new type of construction robot task planning system that connects BIM and ROS. The potential benefits include the utilization of rich infrastructures of robotics industry such as robot shape and functional representations, motion planning algorithms, physics-based simulations, and sensor simulation tools, for construction robot task planning.

- H. Kim, K. Anderson, S. Lee, J. Hildreth, Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology, Autom. Constr. 35 (2013) 285–295, https://doi.org/ 10.1016/j.autcon.2013.05.020.

  Kim et al. proposed a new workflow of producing a construction schedule using model information in BIM, predefined sequencing rules, and task productivity rates.

- K. Kim, J.W. Park, C. Cho, Framework for automated generation of constructible steel erection sequences using structural information of static indeterminacy variation in BIM, KSCE J. Civ. Eng. (2020), https://doi.org/10.1007/s12205-020- 0163-6.

  However, no method properly utilized relationships between building elements to robustly generate 4D BIMs that are structural stable during installation. This research presents an approach to the automated generation of structurally stable construction sequences using a 3D BIM. Focusing on steel erection, we create a framework integrating a 3D BIM and algorithms to create a 4D BIM with detailed steel erection sequences of individual elements. This research ex-

plores an approach to a variation of static indeterminacy for each installation process of steel elements.