A Framework Architecture for the Information Backbone for Robotic Construction

Tamira Wrabel[1,2] and Sebastian Esser[1,2] André Borrmann[1,2]

[1]School of Engineering and Design, Technical University of Munich, Germany
[2]Georg Nemetschek Institute, Technical University of Munich, Germany

tamira.wrabel@tum.de, sebastian.esser@tum.de, andre.borrmann@tum.de

Abstract -

This paper introduces an orchestration framework for the Information Backbone for Robotic Construction (IBRC), a unified platform that connects fragmented planning and on-site robotic construction workflows. Construction sites require continuous coordination among heterogeneous robots, humans, machines, and digital models, yet existing solutions typically address isolated integrations (e.g., BIM-to-robot pipelines or ROS-based execution) without an end-to-end backbone. IBRC structures product–process information through staged refinement from Building Information Modeling (BIM) to a Fabrication Information Model (FIM) and a Robot Execution Model (REM). The conceptualized framework is implemented as a microservice architecture, providing maximum modularity and flexibility. We demonstrate feasibility with an end-to-end prototype that derives robot-executable tasks from BIM and dispatches them to a robot simulation with runtime feedback.

Keywords -

Robotics, BIM, Microservices, REM, Kafka, Robotic Construction, Multiple Robot Systems

1 Introduction

The digital evolution over the past years, with developments such as Building Information Modeling (BIM) and Digital Twins (DT), has primarily focused on design and planning activities for building construction [1] [2]. However, the on-site construction processes remain largely manual and fragmented, thereby limiting the potential for automation. In contrast, other industrial sectors such as automotive manufacturing and logistics have successfully leveraged automation and robotics for efficiency, precision, and cost savings [3]. Here, robots execute one or a limited number of well-defined tasks repeatedly within highly controlled environments, such as a manufacturing facility.

Transferring robotic approaches commonly established in other industry branches into the construction domain poses significant challenges, as a construction environment is unstructured, unpredictable, and dynamic. Every construction site is different and requires individual building processes. Furthermore, any built asset is considered a unique item, which further hinders the application, as necessary processes must be tuned individually to meet the specific conditions present on a construction site. The deployment of autonomous robotic agents under such conditions requires not only mechanical and sensory adaptation, but above all, a robust digital coordination infrastructure that controls the entire process from planning to execution [4].

A unified coordination and collaboration platform can solve this need for real-time, modular integration between planning tools and heterogeneous robots [4]. Current similar approaches can be found in the field of Digital Twins (DT) [5] [6]. However, those are mostly designed to support the current construction process and do not include robotic fabrication. For robotic construction, there is a dedicated need for a holistic orchestration of task planning, allocation, and execution, as well as multi-robot coordination, human-robot communication, status monitoring, and robot-robot interaction. While several partial solutions already exist, such as ROS-based task execution [7] or BIM-to-robot [8] [4] pipelines, these are often limited to certain construction processes, environments, or robot types. Overall, the construction industry requires a paradigm shift in robotic construction. However, robotic fabrication remains expensive, which makes it essential to test construction processes in simulation before deploying them on-site. Nonetheless, since construction sites often face unforeseen changes, the system must be capable of adapting to those.

This paper introduces the Information Backbone for Robotic Construction (IBRC) (Figure 1) as a central platform that combines planning, simulation, and on-site execution. IBRC serves as a shared source of truth for decisions made by humans and robots, supporting a gradual refinement of product-process information from the Building Information Model (BIM) to the Fabrication Information Model (FIM) to the Robotic Execution Model (REM).

In detail, this paper aims to approach the following research questions:

- RQ1: Which conceptual framework is required to

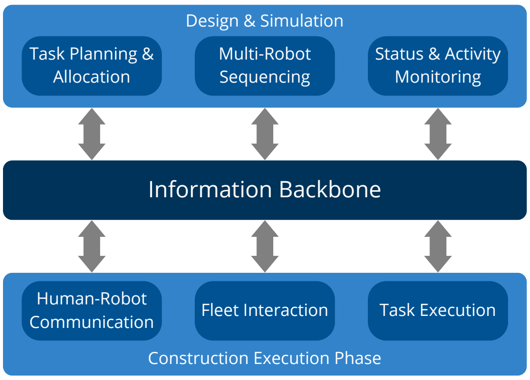

Figure 1. Architectural Framework of the Information Backbone for Robotic Construction

capture the requirements of on-site robotic construction, in particular for representing and coordinating robots, construction methods, sensors, human actors, and digital models at appropriate levels of granularity?

- RQ2: How should a system architecture instantiating this framework be designed to enable scalable integration and autonomous coordination of heterogeneous robotic and digital systems across on-site and off-site construction workflows?

To avoid ambiguity, we use the following terminology throughout the paper. Terminology: *IBRC* refers to the overall concept; the *Backbone* is a concrete platform instance providing shared data, APIs, and event streams; the *framework* describes the BIM→FIM→REM refinement logic; and the *system architecture* denotes the microservice decomposition and communication stack.

In summary, this paper contributes (1) the IBRC conceptual framework that structures robotic construction information into BIM, FIM, and REM, (2) a microservice reference architecture that instantiates it, and (3) a prototype connecting a BIM viewer to a robot execution simulator via an event-driven pipeline.

2 Background

2.1 Reference System Architecture from Digital Twin platforms

Autonomous robotic construction systems require robust, scalable, modular, and maintainable software architectures that support heterogeneous communication and coordination effectively [4]. Similar architectural approaches are evident in the field of Digital Twins. A reference architecture for Digital Twin Construction (DTC) [6] presented a process-oriented schema designed to capture the core concepts of current construction projects, demonstrating how to integrate data and information management from multiple sources within a single platform.

In robotic construction, multiple data sources, components, and processes are involved, which are constantly changing and differ for each project [4]. Therefore, the architecture needs to be adaptable, and the task of each component needs to be clearly defined. This highly relates to the primary principle guiding the design of microservices, which is the single-responsibility principle. Consequently, each service only fulfills one task, but it does so very well [9]. This enables the development of scalable, interchangeable architectures.

For example, the OpenTwins project, presented by [5], is based on such a microservice architecture. The platform is designed to maximize modularity, scalability, and reusability. It offers options for Data Storage, real-time streaming, and 3D visualization. With their approach, the authors showcased an open-source DT platform that is able to visualize and modify data. However, both of those approaches, DTC and OpenTwins, are limited to construction sites that do not involve robotic construction. As robotics extends those frameworks, it presents additional challenges, such as interaction between robots and distributed decision-making, task planning, and allocation, that need to be tackled.

2.2 Robotic Construction using BIM

Approaches for Robotic fabrication on construction sites have already been explored in several papers. These concepts address parts of the overall idea of integrating robots, but do not yet encompass the entire construction process, from planning to execution. Approaches such as RCAN [8] and CONCERT [7] address the topic of integrating Robotic Construction using BIM.

The Robot Construction Action Node (RCAN) [8] connects BIM-based smart components with the Robotic Operating System (ROS) for flexible construction planning and execution. It focuses on the exchange of information between BIM and a robot, composing a construction process from these nodes and applying it to a specific construction project. Their approach demonstrated three basic examples involving construction activities, such as picking, transporting, moving, and installing. However, they have only tested it in a simulation environment with a single robot so far.

A fully reconfigurable, modular collaborative robot, called CONCERT [7], has been introduced. This robot is capable of performing a wide range of tasks with high-power density motors and modularity, enabling it to perform various tasks. The validation was conducted across diverse use cases, such as drilling, sanding, plastering, and collaborative transportation. Their project demonstrates a capable robot that can be employed for various on-site tasks; however, they do not offer an overarching coordination framework necessary to manage this robot alongside multiple other robots on-site.

Through these and other projects, BIM was identified as a solid base for construction as it provides the necessary spatial, semantic, and process-related information in a structured format [10]. But to fully enable robotic construction, the BIM information needs further enrichment and refinement. Furthermore, an ongoing challenge is to connect those approaches within a central platform to enable overall process control.

2.3 Communication Management

An essential feature of such a platform is the ability to exchange data between components to receive tasks, update status, plan resources, and allocate robots. Implementing architectures like the previously referenced OpenTwins project necessitates a robust framework for communication and data exchange. For their project, Robles et al. utilized, among other tools, message streaming and application programming interfaces (APIs) [5]. Another approach is the Internet of Robots [11], which introduces an open-source cloud robotic framework that enables the management, control, and monitoring of robots over the internet. Addressing issues such as efficient management of users and robots through the web by utilizing WebSockets to enable real-time, bidirectional communication between robots and users. This approach showcases how the connection between users and robots can be handled through the internet. However, it is limited to robot-human communication and has not been designed for remote construction sites.

All of these previously mentioned approaches are solutions for specific problems, but they do not consider the overall construction process from planning to robot construction with all involved actors and processes. The BIM-to-robot pipeline focuses solely on solving this for singular robots and does not introduce an overall platform. The advantage of the IBRC is that it combines this pipeline with a holistic concept working for simulation and real-world construction.

3 Information Backbone for Robotic Construction

As already outlined, the unique character of construction projects and the dynamic, constantly evolving nature of outdoor construction sites pose challenges throughout the entire process, from online planning and simulation to anticipating and adapting to execution on-site. In this paper, we explicitly distinguish between *planning* and *execution*. Planning refers to off-site (or pre-runtime) activities that refine and enrich product-process information from BIM to FIM and REM and synthesize executable tasks. Execution refers to run-time, on-site (or simulated) task dispatch, robot-side control, monitoring, and feedback. The *Information Backbone of Robotic Construction* combines both parts, offering a platform for managing, controlling, and executing robotic construction as a shared source of truth.

3.1 Framework

The following paragraph provides a detailed description of the Information Backbone for Robotic Construction, divided into two phases: Planning and Execution. The product-process information managed by the Backbone comprises three main stages: BIM, FIM, and REM, as illustrated in the upper part of Figure 2. The planning stage relies on a Building Information Model (BIM) that represents the designed building. The second step involves an enriched BIM with fabrication-level information (Fabrication Information Model, FIM, [12]), which specifies constructible elements, methods, and sequencing. And third, a Robot Execution Model (REM), which encodes robot-specific tasks and parameters for on-site execution or off-site simulation.

3.1.1 Planning: BIM-FIM-REM refinement

The process is initiated from a design-oriented BIM model that encodes the building's geometry and composition at the level of building elements (e.g., walls, slabs, doors, and beams), along with their spatial, functional, and logical relationships. For robotic construction, however, this representation is typically too coarse, as robots require task descriptions at a finer, fabrication-level granularity (e.g., individual components and operations rather than aggregate building elements). Such execution-oriented detail is typically not included in the design (BIM) model and must be derived and enriched during the planning of robotic execution. This involves deciding on a construction method for realizing the building elements using robotic systems. Typical examples for producing a wall are: Brick-laying, In-situ concreting, or Additive Manufacturing.

As part of the BIM-FIM process, model elements must be further decomposed to represent each atomic component explicitly. This is performed in the first step of the *assembly2part* process. The result is the Fabrication Information Model (FIM), which identifies the individual parts of a building element and enriches them with fabrication-level information. While the FIM describes *what* has to be fabricated, it does not yet specify *how* robots should execute these operations. This is addressed in the subsequent *part2RobotAction* process, where the FIM is augmented

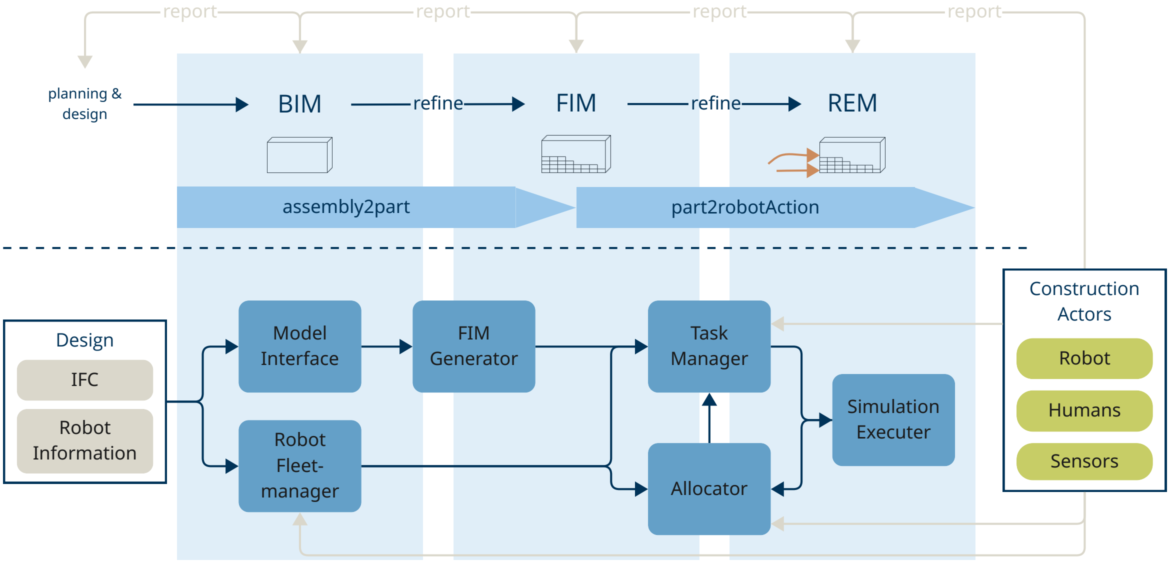

Figure 2. The planning part of the Information Backbone for Robotic Construction supporting the refinement process from BIM to FIM to REM. The different representation layers comprise specific services that provide thorough methods for planning and steering.

with robot-specific actions that define the paths and operations required to assemble the elements. This includes, for example, trajectory information that enables robots to approach the parts and place them at their designated locations. The resulting representation is the Robot Execution Model (REM), which is explicitly tailored to robot operation.

3.1.2 Execution: task dispatch, autonomy, and feedback

The execution phase starts once REM-level tasks are dispatched to robotic agents. While the backbone provides task intent, parameters, and reconditions, robots preserve local autonomy: they execute tasks using their internal controllers and react to on-site conditions (e.g., obstacle avoidance). During execution, robots, sensors, and human operators continuously provide status and observations back to the backbone, enabling monitoring and triggering replanning when deviations occur.

3.1.3 Complexity of Framework

Within the framework, substantial information must be processed, reviewed, enriched, and reconciled across multiple components. However, the two transformation processes *assembly2part* and *part2RobotAction* should not be overloaded with planning decisions. Their primary role is *model refinement*: *assembly2part* decomposes BIM elements into fabricable parts and enriches them with fabrication-relevant attributes (FIM), while *part2RobotAction* augments these parts with *candidate* robot actions, constraints, and execution-relevant parameters (REM). Decisions such as robot assignment, task ordering, and start times are computed by a dedicated planning layer (e.g., allocation and scheduling services) that evaluates robot capabilities and availability and, where necessary, uses simulation to validate feasibility and resolve conflicts before dispatching tasks. Consequently, BIM-to-FIM-to-REM is not a linear pipeline where each step is executed only once. Instead, it is an iterative process: planning and execution continuously produce status reports and deviations that trigger selective updates and replanning, which is reflected by the feedback loops (light grey arrows) in Figure 2.

3.2 System Architecture

Building on the BIM-FIM-REM refinement framework, we design a microservice-based backbone that enables the modular integration of planning and execution components, supporting heterogeneous robots, tools, and data sources. The goal is to create a scalable platform that can be extended per project while keeping responsibilities separate and communication decoupled. To avoid mixing responsibilities, we group services by phase: (1) planning services that derive models and synthesize tasks, and (2) execution and monitoring services that connect to robot fleets, collect feedback, and expose runtime state.

Extensibility is extremely important, as it enables the addition of more robots, components, and other systems, allowing the Backbone to be adaptable to the individual requirements of construction projects. Furthermore, scalability is an important factor, allowing scaling to the appropriate size of the project. Recognizing that robotic fabrication within construction projects necessitates simulating workflows beforehand to facilitate fleet management and resource allocation in a cost-effective scenario.

The proposed system architecture comprises six core components: the human operator, the user interface, web services, the robotic agents, sensors, and data storage systems. Figure 3 illustrates these parts and their connections. The user interface features a building model viewer that allows users to visualize and interact with the digital representation of the construction environment. The interface is connected with the services, which execute and provide the information and computing. Each web service is developed independently, resulting in a highly decoupled structure that enables individual deployment and extension of the architecture as needed. Through a dedicated Communication Stack, services and components exchange messages, send feedback, interact, store data, and receive information.

Specific system functionalities of the Backbone are delegated to dedicated web services. This modularization enhances system scalability, promotes separation of concerns, and facilitates parallel development by enabling multiple contributors to work independently on distinct components. Each microservice in the backbone is designed with a single, clearly defined responsibility [9]. The following describes an initial suggestion of services shown in Figure 3. However, this system will be expanded as needed.

3.2.1 User Interface

The Backbone will be accessible through the User Interface, which provides the frontend for the other services. A central component of the user interface is a *Model-Viewer* that allows users to interact with the building model at various levels of granularity. The viewer will support both static inspection (e.g., design phase) and dynamic visualization linked to task planning and execution. Additionally, a specialized *robot planning viewer* will display either the robot simulation results or the robot's next planned actions in real time. This visualization is designed to support human supervisors in monitoring progress, intervening as necessary, and verifying the robot's actions against the construction model.

3.2.2 Planning services

In this phase, the human operator is responsible for planning and controlling the construction process. The "Model Provider"-Service handles everything related to model upload, validation, versioning, and conversions, storing raw files in object storage and metadata in a database. The "FIM Generator"-Service consumes the uploaded model data and builds a queryable projection, featuring spatial queries, relationships, and assignment of methods for construction elements, explained in more detail in [13]. The "Task Manager"-Service generates decomposed executable activities based on the previously generated graph, containing construction steps.

3.2.3 Execution services and monitoring

The "Fleet Registry"-Service enables the registration of new robots for the construction project, along with associated vendor fact sheets (including specifications and capabilities) and URDF (Unified Robot Description Format) models, allowing other services to retrieve consistent, up-to-date robot information. Furthermore, the fleet registry keeps track of the availability of the robots on site, so whether they are online, available, and their battery status. The "Allocator"-Service receives information from the "Task Manager" about incoming tasks, the simulation, and the robots. It matches tasks with resources, and the result is provided to the simulation, as well as the "Task Manager" and the "Data Collector". The "Data Collector" collects various types of data, including sensor or simulation results, and matches them with environmental information, such as sensor data, providing it to the user interface.

3.2.4 Communication Stack

As shown in Figure 3, the services need to communicate with each other. The communication stack of the Backbone is structured into three distinct layers. At the highest level, the Backbone Communication is implemented using a REST API, which enables communication between external users and microservices to the Backbone. The middle layer, Backbone-Robot Communication, is responsible for exchanging messages between the Backbone and the Robot-Network, as well as other external components, such as sensors, to assign tasks to the robots and receive feedback. This layer is realized using message streaming. In our prototype, this streaming layer is instantiated using Apache Kafka [1], where the backbone publishes task intents and receives execution feedback via dedicated topics (4.2). However, this workflow does not depend on Kafka; it can also be replaced with a similar message broker such as

[1] https://kafka.apache.org/

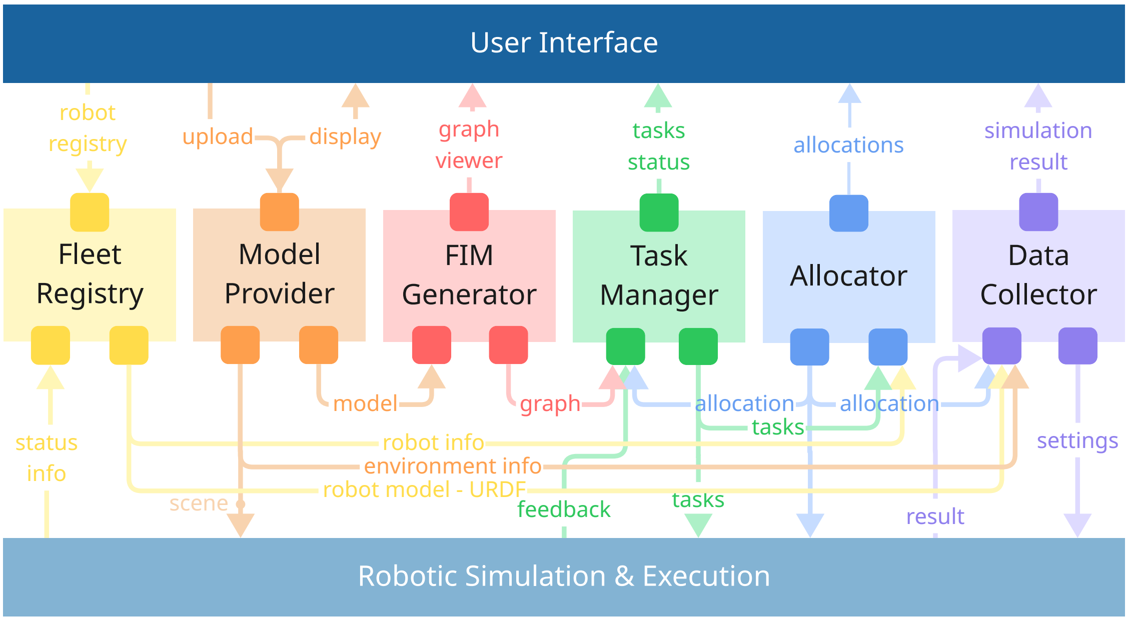

Figure 3. Overview of the Service Architecture of the Information Backbone Implementation

MQTT ([2]). At the lowest layer, the Robot-Network Communication handles communication between the robots and their immediate network environment. This can be achieved via a local mesh network.

4 Application Scenario

We demonstrate the framework and architecture in an end-to-end prototype that transforms an uploaded IFC model into robot-executable tasks and dispatches them to a ROS2-based simulation. This example demonstrates the abstract bridging processes *assembly2part* and *part2RobotAction*, introduced in Section 3.1, implemented in a concrete prototype. It was evaluated against the following four main criteria. First, the ability to integrate new robot types and add new services to the system, for example, by introducing an ”Analytics”-Service that evaluates task status. Furthermore, we demonstrate that we can interchange the uploaded model without requiring changes to the services, while still sending tasks to the robots. Additionally, the structure enables monitoring and receiving feedback from all endpoints (robots, humans, sensors), as well as visualizing them within a single, combined interface. Lastly, we can send tasks to any type of robot, as they are sent via a message broker and are not bound to a specific robot operating system. The architecture's components are illustrated in Figure 3, comprising three stages of the backbone: the User Interface, the Services, and the robotic simulation. The system is deployed using Docker containers, allowing dynamic execution and scaling.

4.1 Planning phase

The User Interface (UI) provides a unified frontend interface built with React [3] and a web viewer based on the Xeokit framework [4]. From this interface, users can trigger the complete BIM-FIM-REM pipeline. The IFC viewer is connected to the ”Model Provider”, which facilitates uploading, storing, and loading IFC models into the backbone. Upon availability of a model, the ”FIM Generator” derives fabrication elements from the IFC data. Construction methods can be assigned or updated on IFC elements within the viewer; these changes are propagated to the graph via the ”FIM Generator”, thereby updating the FIM representation without requiring manual database interaction [13]. The FIM alongside the IFC-Model is shown in Figure 4. After the FIM representation has been generated, the ”Task Manager” invokes a task worker that computes activities from the graph and returns a set of robot-executable tasks, published through Kafka.

[2] https://mqtt.org/

[3] https://react.dev/

[4] https://xeokit.io/

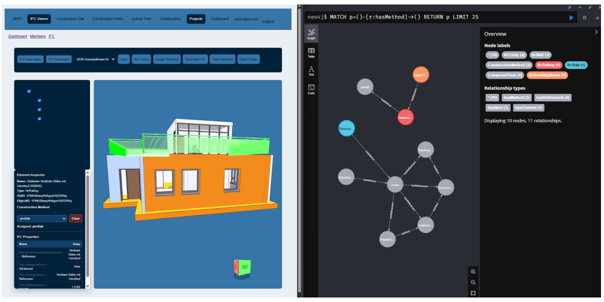

Figure 4. Insight in Frontend-Viewer alongside with the FIM representation

4.2 Online Control in Execution: Backbone-Robot Bridge

In the execution phase, a ROS2 node subscribes to the task topic, translates backbone messages into robot commands, and drives the simulation/execution environment. Runtime status and feedback are propagated back through the same bridge, allowing the UI to visualize the live execution state and the backbone to trigger replanning if required.

We use Kafka to decouple runtime execution from planning and to avoid tight coupling between the backbone and heterogeneous robot stacks. Task dispatch is handled via a `tasks` topic, while runtime feedback is returned via a `status` topic. Messages carry a unique `task_id` and link execution events to FIM entities. Those tasks generated by the "Task Manager" are forwarded to a dedicated Kafka topic. Then the tasks are consumed by robot-side clients (consumer groups) and translated to ROS2 commands. Status events follow a simple state model (`queued/accepted/executing/completed/failed`) and optionally include progress and error codes. Status updates ensure that the planning interface reflects the live execution state of the corresponding model elements.

5 Discussion and Conclusion

This paper aimed to address two questions: (RQ1) What conceptual framework is required to capture the requirements of on-site robotic construction, and (RQ2) How should a corresponding system architecture be designed to enable the scalable integration and autonomous coordination of heterogeneous robotic and digital systems? The framework (RQ1) is realized through staged refinement from BIM to FIM to REM (Section 3.1), enabling consistent linking of model elements, fabrication parts, and executable tasks.

Regarding RQ2, we implemented this framework as a distributed, microservice-based architecture. The "Model Provider", "FIM Generator", and "Task Manager" services are connected via REST APIs and an event-driven exchange using Kafka. In this setup, the backbone specifies what is to be done (task and preconditions), whereas robots retain local autonomy in how they execute these tasks and react to on-site conditions. The backbone delivers the task and core requirements, and the robots can autonomously decide locally how to fulfill the task and react to the on-site conditions. The prototype demonstrates that this architecture can establish an end-to-end pipeline from planning tools to construction robots, including task generation, distribution, and feedback.

To provide initial feasibility test of this approach, we implemented a minimal end-to-end prototype that connects a web-based planning interface and a viewer to the IBRC services and a simulated robot. As such, it provides evidence that the conceptual framework is expressive enough to support an end-to-end BIM-FIM-REM pipeline and that the reference architecture can realize this pipeline in a modular, event-driven manner. However, it does not yet constitute a comprehensive validation under real-world conditions. Future work will therefore extend the backbone to support multi-robot collaboration, including task allocation, reallocation, and joint execution. The geometric processing capability will be enhanced to facilitate faster processing and adaptability to various data formats. The architecture will be validated on real projects, both on-site and off-site, with multiple system partners.

Overall, the results suggest that an information backbone combining shared semantics with event-driven exchange is a promising basis for integrating diverse robotic and digital systems at scale and coordinating them without tight coupling. By providing a bridge from model to action (BIM $\rightarrow$ FIM $\rightarrow$ REM) that is not tied to a particular robot or software stack, the IBRC offers a viable starting point for more modular, scalable, and autonomous robotic construction workflows and a foundation on which subsequent work can build more specialized planning, control, and analytics capabilities.

6 Acknowledgments

We gratefully acknowledge the support of the German Research Foundation (DFG) for funding the project under grant FOR 5672, subproject A "Generic framework for simulating and executing robotized construction", grant number 517965147.

Furthermore, we acknowledge the support of the TUM Innovation Network "CoConstruct", the DFG Transregio 277 "Additive Manufacturing for Construction" (grant 414265976), and the EU Pathfinder project "AM2PM – Additive to Predictive Manufacturing for Multistorey Construction using Learning-by-Printing and Networked Robotics"

References

[1] Charles M Eastman. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, 2011.

[2] André Borrmann, Markus König, Christian Koch, and Jakob Beetz. *Building Information Modeling: Technologische Grundlagen und industrielle Praxis | SpringerLink*. VDI-Buch. Springer Vieweg Wiesbaden, 3 edition, 2021. ISBN 978-3-658-33360-7. URL https://link.springer.com/book/10.1007/978-3-658-33361-4.

[3] Wuhui Chen, Yuichi Yaguchi, Keitaro Naruse, Yutaka Watanobe, Keita Nakamura, and Jun Ogawa. A Study of Robotic Cooperation in Cloud Robotics: Architecture and Challenges. *IEEE Access*, 6:36662–36682, 7 2018. ISSN 21693536. doi:10.1109/ACCESS.2018.2852295.

[4] Hongzhe Yue, Qian Wang, Zixuan Zhao, Sha Lai, and Guanying Huang. Interactions between bim and robotics: Towards intelligent construction engineering and management. *Computers in Industry*, 169:104299, 8 2025. ISSN 0166-3615. doi:10.1016/J.COMPIND.2025.104299. URL https://www.sciencedirect.com/science/article/pii/S0166361525000648?utm_source=chatgpt.com.

[5] Julia Robles, Cristian Martín, and Manuel Díaz. Opentwins: An open-source framework for the development of next-gen compositional digital twins. *Computers in Industry*, 152:104007, 11 2023. ISSN 0166-3615. doi:10.1016/J.COMPIND.2023.104007. microservice architecture.

[6] Jonas Schlenger, Kacper Pluta, Alwyn Mathew, Timson Yeung, Rafael Sacks, and André Borrmann. Reference architecture and ontology framework for digital twin construction. *Automation in Construction*, 174:106111, 6 2025. ISSN 0926-5805. doi:10.1016/J.AUTCON.2025.106111. URL https://www.sciencedirect.com/science/article/pii/S0926580525001517. distributed vs centralized architecture.

[7] Luca Rossini, Edoardo Romiti, Arturo Laurenzi, Francesco Ruscelli, Marco Ruzzon, Luca Covizzi, Lorenzo Baccelliere, Stefano Carrozzo, Michael Terzer, Marco Magri, Carlo Morganti, Maolin Lei, Liana Bertoni, Diego Vedelago, Corrado Burchielli, Stefano Cordasco, Luca Muratore, Andrea Giusti, and Nikos Tsagarakis. Concert: a modular reconfigurable robot for construction. *Journal of Field Robotics*, 4 2025. doi:10.1002/rob.70092. URL http://dx.doi.org/10.1002/rob.70092.

[8] Aiyu Zhu, Pieter Pauwels, Elena Torta, Hong Zhang, and Bauke De Vries. Data linking and interaction between bim and robotic operating system (ros) for flexible construction planning. *Automation in Construction*, 163:105426, 7 2024. ISSN 0926-5805. doi:10.1016/J.AUTCON.2024.105426.

[9] Sam Newman. *Building Microservices, 2nd Edition*. O'Reilly Media, Inc., 2 edition, 2021. ISBN 978-1-4920-3401-8. URL https://learning.oreilly.com/library/view/building-microservices-2nd/9781492034018/.

[10] Pieter Pauwels, Rens de Koning, Bob Hendrikx, and Elena Torta. Live semantic data from building digital twins for robot navigation: Overview of data transfer methods. *Advanced Engineering Informatics*, 56:101959, 4 2023. ISSN 1474-0346. doi:10.1016/J.AEI.2023.101959.

[11] Fabeha Raheel, Laibah Raheel, Hassan Mehmood, Muhammad Bilal Kadri, and Umar Shahbaz Khan. Internet of Robots: An Open-source Framework for Cloud-Based Global Remote Control and Monitoring of Robots. In *2024 International Conference on Robotics and Automation in Industry (ICRAI)*, pages 1–6, 2024. doi:10.1109/ICRAI62391.2024.10894149. URL https://ieeexplore.ieee.org/document/10894149.

[12] M. Slepicka and A. Borrmann. Fabrication Information Modeling for Closed-Loop Design and Quality Improvement in Additive Manufacturing for Construction. *Automation in Construction*, 168: 10579, 2024. doi:10.1016/j.autcon.2024.105792. URL https://www.sciencedirect.com/science/article/pii/S0926580524005284?via%3Dihub.

[13] Maikel Brinkhoff. An integrated product–process representation for deriving robotic task structures using graph transformations. In *ISARC*, 2026. Manuscript submitted, under Review.