The information backbone of robotized construction: Principles, procedures, and software architecture

A. Borrmann[1], T. Bruckmann[2], K. Dörfler[3], T. Hartmann[4], K. Smarsly[5], M. Brinkhoff[1], A. Das[3], S. Esser[1], M. Hassaan[1], C. Jeziorek[2], N. Kim[1], M.R. Kolani[1], A. Maurya[4], S. Nousias[1], P. Petropoulakis[1], A. Tandon[5], T. Wrabel[1]

[1]Computing in Civil and Building Engineering, Technical University of Munich, Germany
[2]Chair of Mechatronics, University of Duisburg-Essen, Germany
[3]Digital Fabrication, Technical University of Munich, Germany
[4]Civil Systems Engineering, Technical University of Berlin, Germany
[5]Digital and Autonomous Construction, Hamburg University of Technology, Germany

andre.borrmann@tum.de

Abstract -
Advances in robotic systems, sensing technologies, and artificial intelligence offer substantial potential to automate construction processes. However, current construction robotics applications are limited to specific tasks in isolated systems, lacking the integration required for coordinated, large-scale robotized construction that is applicable in a wide spectrum of trades. A key challenge is the absence of digital frameworks integrating product information, construction processes, robotic capabilities, and execution feedback across planning, simulation, and on-site operation. This paper presents the concept, principles, and software architecture of the information backbone of robotized construction (IBRC), a unifying framework for planning, simulating, and executing construction projects using coordinated, heterogeneous robotic systems. The IBRC comprises two tightly coupled environments: an offline planning environment and an online execution environment. The planning environment supports successive refinement of product and process information, starting from a building information model (BIM), via a fabrication information model (FIM), to a robot execution model (REM). The refinement is realized through a graph-based product–process representation and rule-based construction recipes, enabling the derivation of atomic construction actions at a granularity suitable for robotic execution. The execution environment builds on the results and supports both simulated and real-world operation. It relies on middleware, such as ROS 2 for real-time robot communication or Apache Kafka for scalable integration with higher-level services. Central orchestration is achieved through behavior-tree-based multi-robot coordination.

Keywords -

Robotized Construction; Information Backbone; Construction Robotics; BIM-based Process Planning; Multi-Robot Coordination; Fabrication Information Model

1 Introduction

Recent advancements in mechatronics, sensor systems, and artificial intelligence have substantially enhanced robotic capabilities, revolutionizing industrial manufacturing and increasing automation across many industries. It is, therefore, not surprising that robotic systems have made significant advancements in construction environments over the past years [1]. However, the robotic automation in construction thus far has been limited to isolated systems and highly specialized processes. Hence, a paradigm shift towards realizing a comprehensive system-of-systems approach is required that facilitates heterogeneous fleets of autonomous and semi-autonomous robotic systems to collaboratively work alongside humans in a well-coordinated manner on-site [2]. Furthermore, to achieve reliable robotized construction, adequate and precise information about the buildings under construction, the construction methods, processes and progress, the position, action, and interaction of robots, as well as the status of the site as a whole, must be available in a coherent digital representation. On the basis of the information, the central coordination of all on-site robotic activities must be established. Today, appropriate information backbones for simulating and executing robotized construction processes do not yet exist. Therefore, this paper proposes the concept of an information backbone of robotized construction (IBRC) that provides these functionalities [3].

The IBRC aims to provide a framework that enables a virtual laboratory environment where new robotized construction methods can be tested and developed. The framework provides fundamental services, such as digital representations of the facility under construction and the involved processes, and, at the same time, provides flexible interfaces to allow the integration of any type of robotic system, sensors, and construction methods. This paper presents the general concepts underlying the IRBC. To this end, rather than describing isolated applications, robotized construction is framed as a two-phase workflow, encompassing (1) the configuration and planning of a largely robotized construction project and (2) its simulation and on-site execution through the orchestration of a fleet of heterogeneous robots across diverse construction tasks.

Regarding (1), the paper defines a digital workflow in which product and process information are represented as interdependent graph data structures. These graph representations are co-evolved and mutually constrained, enabling the consistent formulation of constructability, sequencing, and system capabilities. Most importantly, the workflow results in a set of atomic, executable actions that can be assigned to robotic systems. Regarding (2), the simulation environments, communication protocols, and interactions with a central coordination system are discussed.

2 State of the art

A crucial aspect of construction robot task and motion planning is the effective use of BIM and construction schedules to generate executable plans. In early work on integrating BIM with robotics, Meschini et al. [4] highlighted the need to link BIM with the Robot Operating System (ROS) [5] to support construction automation. Their study demonstrated scenarios such as precast timber module installation and robotic assistance for elderly care, showing how BIM-based models can be used to evaluate alternative robotization strategies. However, it remained at a high level and did not provide a concrete BIM–ROS integration method. Addressing this limitation, Ibrahim et al. [6] proposed mapping the as-built state of an indoor construction site by generating waypoints in BIM, transforming them into robot navigation coordinates, and performing LiDAR-based mapping during execution.

Similarly, Mantha et al. [7] proposed task allocation and route planning for multiple service robots using simplified 2D building representations, reflecting the broader limitation of insufficient detail in construction schedules. In parallel, CAD-based and geometry-driven frameworks such as compas-fab [8] provide more direct links between design models and robotic task planning, simulation, and execution.

According to the process hierarchy introduced by Everett and Slocum [9], manufacturing-oriented robot process planning operates at the level of elemental motions, whereas construction schedules are typically specified only at the level of coarse-grained activities. This mismatch creates a fundamental gap between the information available in construction plans and the level of detail required for robotic execution. Even with BIM, it is impractical for planners to manually define robot-level tasks at this granularity, and recent approaches [10, 11, 6] have not yet closed this gap.

One of the few approaches addressing this issue is Kim et al. [12], who extend BIM with task-planning information to generate motion-level robot behaviors. Their ROS-based prototype for indoor wall painting includes conversion from IFC to a ROS-compliant world model and integrated localization, navigation, and motion planning.

Related work by Zhu et al. [13, 14, 15] focuses on robotic assembly of prefabricated elements. The smart construction object (SCO) concept assigns state and requirements to components to coordinate multiple robots, and subsequent work enables automatic conversion from IFC to simulation description format (SDF) to create smart building components for simulation.

Despite these advances, most approaches remain restricted to comparatively simple assembly scenarios and lack generality. A comprehensive framework capable of planning and executing diverse robotized construction processes in a coordinated manner is therefore still required.

3 The information backbone: Overall architecture and components

The proposed framework for planning and executing robotized construction projects is organized around an information backbone that integrates product and process representations across planning and execution stages. The backbone provides a consistent data model and set of interfaces through which design information, construction logic, and robotic task specifications are incrementally developed, validated, and executed. While the detailed implementation is described in [16], this section outlines the underlying principles, the overall software architecture, and the interaction between the core components of the IBRC.

At the architectural level, the proposed system consists of two tightly coupled environments:

1. The **planning environment** (“offline” components) that allows for the iterative development of the construction project, starting from the overall building design down to precise instructions for individual robotic systems.
2. The **execution environment** (“online” components) that allows for the steering and coordination of a heterogeneous fleet of robotic systems, implementing the assigned tasks to complete the construction project, while being able to react to unforeseen events. The execution environment is designed to ensure a seamless transition between simulation-based validation and execution in physical environments.

Both environments operate on a shared information backbone that encodes the evolving relationship between the construction product and process, and may employ model-based methods and process simulations.

3.1 Planning environment

The planning module of the IBRC enables the successive and parallel refinement of both the product (= building components) and the process representation. The module

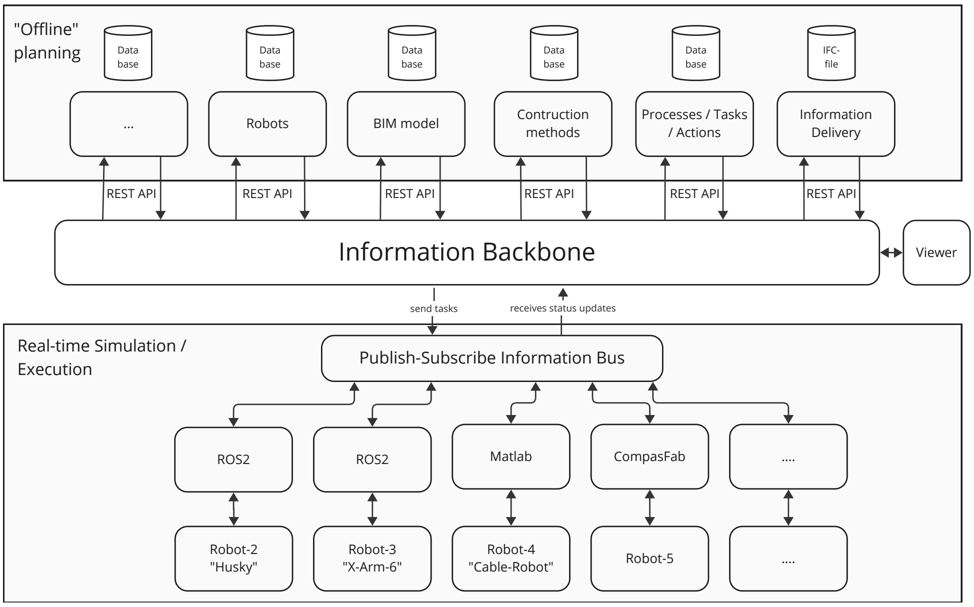

Figure 1. The architecture of the information backbone consists of two major parts: The planning environment and the execution environment

consists of several modules that provide or process data in a service-oriented manner [16].

3.1.1 BIM to FIM to REM: Graph-based refinement of products and processes

The product-process elaboration process is outlined in Fig. 2 and described in detail in [17]. The starting point is a relatively coarse BIM model of the building to be constructed, comparable to an as-designed LOD300 model. The model is represented as a graph that combines two trees, with one tree representing the hierarchical breakdown of the product side and the other representing the hierarchical breakdown of the process side. The process tree is automatically derived from the product graph. Product nodes are linked to the set of process nodes that realize the construction.

By assigning construction methods to individual building components, such as *in-situ concreting, bricklaying,* or *additive manufacturing*, both the product and process graphs are refined. After refinement, the model is referred to as a Fabrication Information Model (FIM) and has a much finer resolution than typical BIM models [18]. FIM contains and represents all atomic – in the sense of indivisible – elements that have to be manipulated for the production of a building element. For example, assigning the construction method *bricklaying* to a wall generates the FIM that contains all individual bricks and the mortar; while assigning the construction method cast-in-situ creates a FIM that represents the concrete, reinforcement, and temporary formwork elements. At the same time, the process graph is further refined and now contains individual activities such as *erect formwork*, *insert reinforcement*, *fill concrete*, and *remove formwork*.

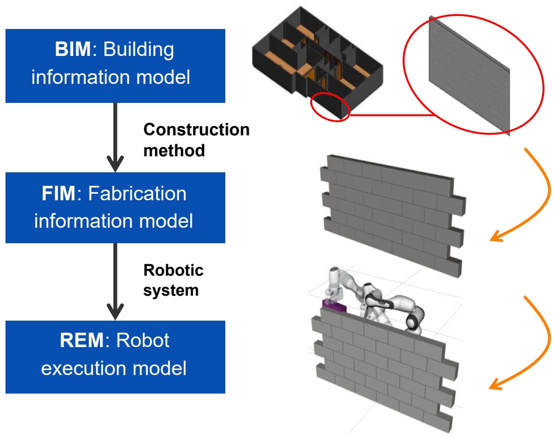

Figure 2. From BIM to FIM to REM

The simultaneous refinement of product and process graph is performed by applying *recipes* that encapsulate the knowledge about specific construction methods [19]. Recipes not only represent the rules for creating sub-nodes for individual process nodes and product nodes, but also define precedence relationships and the required resources (including robots, cranes, construction machines, or human workers). For example, to start "*Fill Concrete*", the activity "*Erect Formwork*" must be completed. For placing formwork and inserting reinforcement, a (robotized) crane is required. The information might later be used e.g. in an event simulation to experiment with the effects that various robot allocation strategies in terms of the amount and type of robots will have on the overall construction schedule, as well as on construction costs.

The product-process graph is refined down to a level where leaf nodes represent atomic elements (e.g., bucket) and atomic actions (e.g., FillBucket), as shown in Fig. 3. Atomic actions are later transformed into manipulation instructions sent to the individual robotic systems. The instructions are described at a level of abstraction that is executable by robotic systems while omitting low-level motion details, allowing a certain degree of interpretation and thus granting robotic systems the autonomy to plan execution. For example, the *transport formwork* action, sent to a robotized crane, specifies only start and end positions, not the exact crane hook trajectory or jib movement.

This kind of fine-grained information – such as trajectory data or actuator commands needed for a certain manipulation task – is represented by a robot execution model (REM). REMs are created in the next step, and typically reside outside the information backbone, in components closer to robot control. While FIM is robot-type-agnostic, REMs are created for specific robot types and contain all set motion data for the respective robot type to complete a task. REM computation employs robot models (of potentially varying granularities) that might be used to e.g. kinematically compute a simple trajectory or even optimize a motion within a dynamical simulation, considering e.g. collisions, motion time or energy consumption. Accordingly, choosing another robot type for the same manipulation task leads to different motions and thus a different REM. Still, the methods and interfaces for REM computation might be generalized: a BIM-FIM-REM toolchain has been demonstrated to generate a REM model for automated masonry with cable-driven parallel robots [20]. Finding a suitable robot involves robot task allocation (section 3.1.3), but nevertheless, there is feedback from the robotic systems on previous steps, as different robots have different characteristics. For example, robots with higher payloads can place larger brick formats.

The Backbone is accessible mainly through APIs, but also a User Interface, consisting of a modular model viewer and a robot planning viewer. The model viewer offers a visual inspection of the building, status tracking, and enables task planning and execution. The robot planning viewer is used to display simulation results and, during execution, the robot's next planned actions in real-time.

3.1.2 Discrete event simulation

The precedence relationships of the process graph define the general sequence of atomic tasks, but sequence alone does not capture the probabilistic time distribution a specific robot needs for each task. Such knowledge about the overall stochastic distribution is, however, important to plan for the resource-intensive robotic equipment and the equally important supply chain that is required to provide the robotic equipment with the required material.

To determine the stochastic duration of overall robotic work processes on-site, we propose to employ constraint-based discrete event simulation [21, 19]. In these simulations, the duration of a directed graph network of tasks is simulated. Constraints determine when a single task in the network can be started: all preceding tasks in the network need to be finished, all required materials need to be available, and all robotic and non-robotic equipment needs to be available. When all prerequisites are met, the simulation starts a task that executes it for a duration drawn from a specific probability distribution representing the stochastic duration of the atomic task, ideally defined based on empirical data collected from previous construction work. During the duration of a task, all resources are bound and not available for starting other tasks.

A stochastic Monte-Carlo repetition of various individual discrete-event simulations then provides important information to strategically plan the site. Variations in the amount and provision of robotic and non-robotic equipment can provide stochastic durations and costs of specific resource allocations. For each provided resource allocation scenario, simulation results provide information about when specific material resources are required, which, in turn, allows for planning material supply chain aspects for the entire construction site.

3.1.3 Robotic skills and task allocation

To enable the selection of suitable entities, such as robotic systems, construction machinery, or humans, for dedicated construction tasks, a formal method for describing the skills, constraints, and requirements is developed. The *skills* section includes an enumeration of tasks such as *lifting, placing, drilling, demolition*, etc. The *constraints* section includes the maximum load as well as the operational range of the system, among others. The *requirements* section includes the necessary space, supply (electricity, material), and the weight of the system to be considered as loads for the underlying structure.

The formal description of robotic skills, constraints, and requirements forms the basis for multi-robot task allocation. Construction tasks derived from the FIM model are assigned to individual robots capable of performing the task. For example, transport before the assembly of a prefabricated component requires a mobile robot capable of grabbing, lifting, and transporting elements of certain sizes and weights. If multiple robots are qualified, they are ranked by performance factors such as speed, required space, etc. The actual task allocation procedure is based on a free-market allocation approach as proposed by [22].

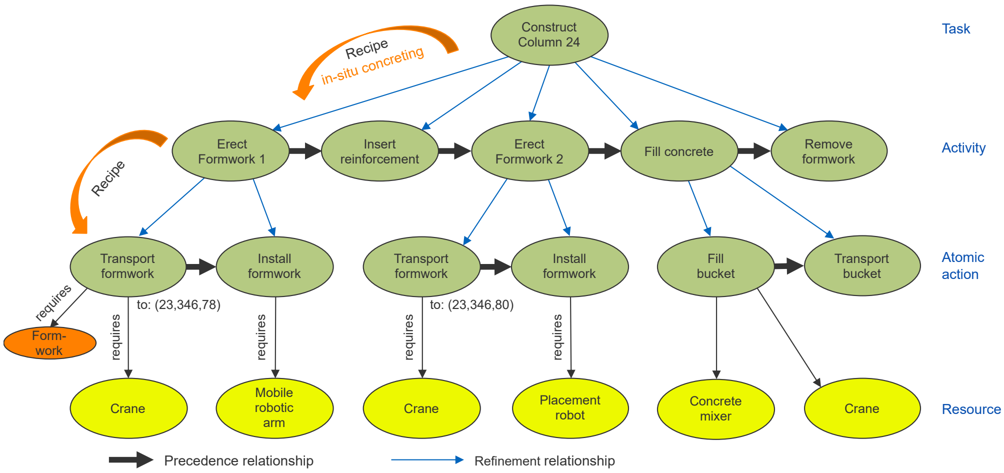

Figure 3. Process graph representing the hierarchical breakdown of the construction processes. The leaf nodes represent instructions for the individual robotic systems.

3.2 Execution environment

The purpose of the second module of the information backbone lies in the execution of the construction processes prepared in the planning environment, primarily involving steering and coordinating the diverse robotic systems. As real-world experimentation is cumbersome and expensive, the proposed approach supports performing robotic simulations with a high level of realism. At the same time, communication protocols are defined to facilitate an easy transition from controlling simulated robots to controlling real robots.

3.2.1 Middleware

Communication between the central coordination module and the robotic systems is handled by middleware, with an implementation using the Robot Operating System 2 (ROS 2) as the primary framework. ROS 2 is a modular, open-source framework for developing, deploying, and operating robotic systems. It provides standardized communication mechanisms, tooling, and abstractions that support distributed, heterogeneous, and real-time-capable robotic applications.

ROS 2 builds on Data Distribution Service (DDS) implementations such as eProsima Fast DDS [1], or Eclipse Cyclone DDS[2], which use the DDS-RTPS wire protocol[3] for message transport over standard network protocols, including UDP and TCP. ROS 2 communication follows a message-oriented paradigm. Data is exchanged through strongly typed messages defined in an interface description language. Messages are published and subscribed via named topics, enabling loosely coupled, asynchronous communication between robot components and processes.

Alternatively, Apache Kafka[4] is available as an additional abstraction layer. While ROS 2 provides a very complete toolset with extensive libraries, computational methods and robot-centric communication, Kafka provides a scalable and fault-tolerant event-streaming that supports integration of external systems such as digital twins, BIM services, or analytics platforms. Bridging ROS topics to Kafka streams allows robot-generated events to be consumed by higher-level coordination and monitoring services, while process-level commands can be reliably disseminated to the robotic fleet. The separation supports heterogeneous system landscapes and enhances scalability without affecting time-critical robot control. Furthermore, Apache Kafka provides an interface to industry-standard bus systems such as EtherCAT. In this context, an interface to MATLAB Simulink is created to enable the simulation of robotic systems outside of the ROS 2 environment.

Exemplarily, this is used for the simulation and control of a cable-driven parallel robot for construction.Due to its size, payload and sensitivity towards stable control frequencies, industrial hard real-time control is required. Accordingly, control is performed on a Simulink target in a Beckhof TwinCAT PLC, while hardware communication is performed via an EtherCAT bus.

3.2.2 Shared model hosting and hierarchical information flow

Beyond communication protocols, the system architecture must define how shared product and process representations are hosted and maintained. In the proposed framework, the BIM and the FIM serve as authoritative,

[1] ttps://github.com/eProsima/Fast-DDS
[2] https://github.com/eclipse-cyclonedds/cyclonedds
[3] https://www.omg.org/spec/DDSI-RTPS/
[4] https://kafka.apache.org/

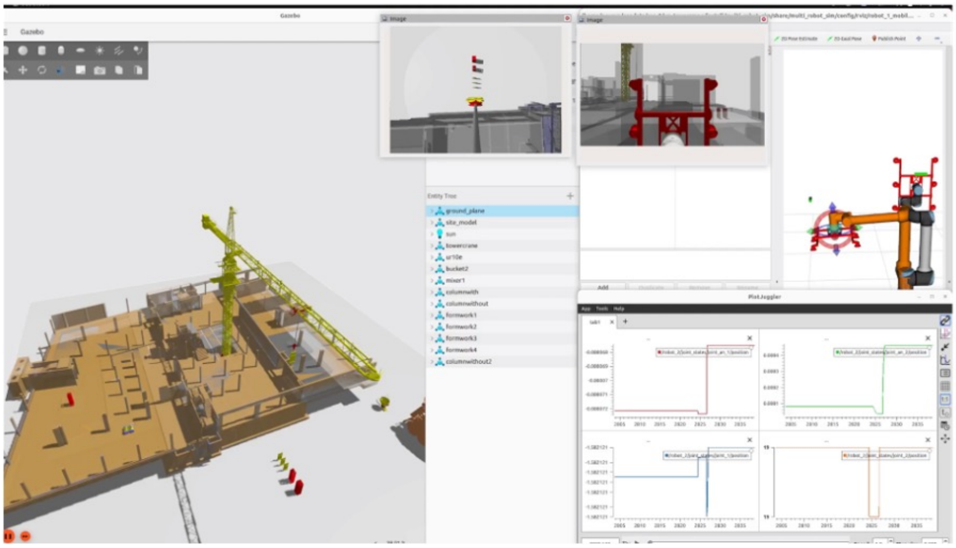

Figure 4. Gazebo is one of the employed simulation environments. Here, the interaction between a robotized crane and a mobile manipulator on a wheeled base is simulated.

global descriptions of the construction state and therefore cannot be treated as local or transient data structures.

Both models are hosted in a cloud environment and are accessible to all distributed agents participating in the construction process, including planning services, simulation components, and robotic execution systems. Shared hosting enables synchronized access to up-to-date product and process information, supporting continuous updates that reflect construction progress, execution outcomes, and detected deviations.

3.2.3 Behavior tree for the central orchestration of the robotic fleet

For executing robotic processes in a coordinated manner, process graphs resulting from the planning environment are transformed into behavior trees (Fig. 5) [17]. Behavior trees are hierarchical, graph-based task control structures for modeling and executing robot behaviors in a modular and reactive manner [23, 24]. The authors decompose complex tasks into reusable nodes, developed in the form of robotic capability subtrees [24], that encapsulate actions, conditions, and control flow, enabling clear separation of decision-making and execution. Due to the scalability, readability, and support for real-time reactivity, behavior trees are widely adopted for robotic task execution and control in both research and industrial applications.

In contrast to controlling individual robotic systems, behavior trees are employed as a centralized coordination mechanism to orchestrate the tasks of an entire heterogeneous robotic fleet. In the IBRC architecture, the leaf nodes of the behavior tree issue messages to the respective robotic systems, enabling coherent and synchronized multi-robot behavior. The messages mirror the level of (process) detail provided by the product-process graph and allocate tasks based on robotic capabilities, leaving a certain freedom of interpretation and degree of independence to the individual robotic systems.

Individual robotic systems may additionally implement local behavior trees for fine-grained control over robotic behavior. In the hierarchical organization of the IBRC, behavior trees do not directly manipulate shared models; instead consume task assignments and constraints derived and report execution results back to the information backbone, enabling closed-loop updates of the global construction state. More details are provided in [17].

3.2.4 Robotic simulation environments

Robotic simulation environments are a central component of the proposed architecture, as construction robots must operate in complex, unstructured, and continuously evolving environments. Reliable interaction with the built environment, human workers, and other robots requires extensive validation of perception, control, and coordination strategies prior to on-site deployment. Simulation therefore provides a controlled setting in which these capabilities can be developed, tested, and refined before being transferred to physical systems.

Within the proposed framework, simulation environments act as first-class execution entities that mirror the physical world within the same software architecture. Simulated robots receive control commands and publish state and sensor data via middleware interfaces, allowing them to function as substitutes for physical robots during planning, validation, and testing (Fig. 4), and enabling seam-

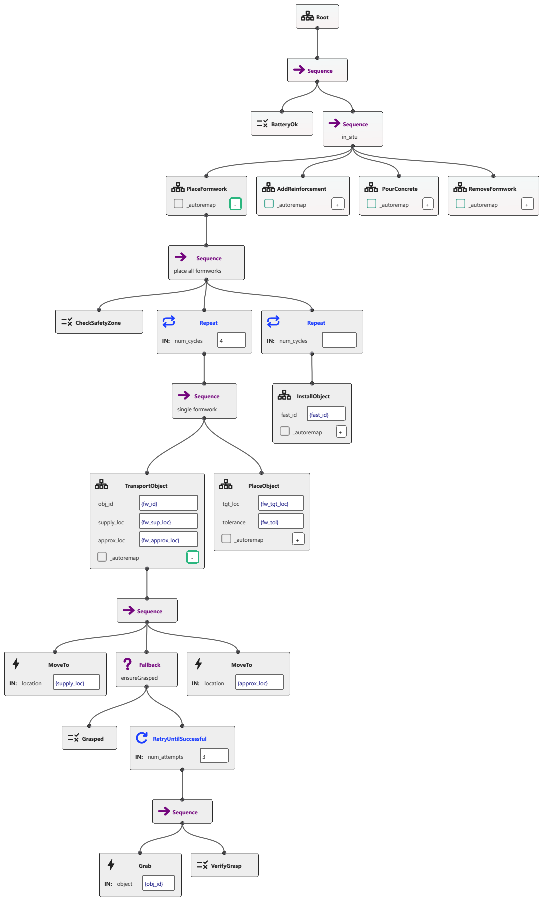

Figure 5. Behavior tree created from the process tree, serving as an executable representation for steering and coordinating the fleet of robotic systems on-site.

less switching between simulated and real execution while preserving identical orchestration and control logic.

Simulation also supports data-driven development workflows. Real-world data acquired through teleoperation, such as camera images, joint states, or encoder readings, can be used to train, calibrate, and refine algorithms that are then evaluated in simulation, which is particularly relevant for construction-specific challenges such as traversing uneven terrain. Core subsystems including controllers, navigation stacks, and SLAM algorithms are therefore developed and validated in simulation prior to deployment. Although simulations cannot fully replicate real-world conditions, noise-aware training and robust algorithm design enable effective transfer to physical robots.

Our framework employs Gazebo, Isaac Sim, and PyBullet as simulation backends, all providing high-fidelity rigid-body physics and tight ROS integration (Fig. 4). Simulation environments are initialized using product and process information from the planning environment. BIM and FIM models are translated into simulator-specific geometry and assets, e.g., SDF entities in Gazebo [25] or NVIDIA-internal formats in Isaac Sim. Robots are instantiated from URDF models provided by the information backbone, allowing the planning layer to remain independent of simulator-specific data structures while supporting multiple simulation platforms. Alternatively, MATLAB Simulink provides powerful tools for modeling construction robots, enabling high-granularity representations of kinematics, dynamics, and physical subsystems such as electronics, pneumatics, and hydraulics.

3.2.5 Human–robot collaboration

Although the proposed framework targets a high degree of robotization, construction remains dependent on human labor and decision-making. Human activities must therefore be explicitly represented in planning and execution, treating the site as a shared workspace where humans and robots perform tasks independently, sequentially, or in close physical interaction.

Human representation is a prerequisite for planning and simulation of collaboration. In simulation, humans can be modeled at different abstraction levels, e.g., kinematic chains, motion-capture trajectories, or avatar-based motion primitives. These models enable analysis of human–robot tasks derived from the process graph and behavior tree, allowing spatial conflicts, interference, and safety-critical situations to be detected before deployment.

Construction activities are classified into robot-only, human-only, and collaborative tasks. Robot-only and human-only tasks can proceed under sequencing constraints, whereas collaborative tasks require coordinated actions. This classification supports task allocation, synchronization, and monitoring across heterogeneous agents.

Coordination is implemented via synchronization constructs embedded in the centralized behavior tree. Robots can wait for human actions, react to changes in human state or position, signal completion, and assist during joint operations. Thus, behavior trees encode both robotic actions and interaction logic for handovers, sequential execution, and parallel work on shared workspaces or workpieces.

Given complementary strengths – robots in repetitive, heavy, and precision-demanding operations; humans in dexterity and situational decision-making – the framework incorporates human-in-the-loop control. Human input can initiate execution, approve transitions, or override robot behavior for supervision and safety.

All human actions and human–robot interactions are reported to the information backbone and integrated into the digital twin, capturing progress, contributions, and interaction states. Simulation further enables validation of collaborative scenarios with respect to safety, efficiency, and usability prior to real-world deployment.

4 Summary

The paper has presented overall concepts, procedures, and the software architecture of the Information Backbone for Robotized Construction. Further details are provided in accompanying papers [16, 17, 25]. The concept has been implemented in a proof-of-concept prototype combining the described modules through a micro-service platform [16]. In the next step, a comprehensive simulation experiment of a close-to-real construction project will be performed to further refine and elaborate the components.

Acknowledgments

The authors gratefully acknowledge the support provided by the German Research Foundation (DFG) for funding the research unit under grant FOR 5672 (grant number 517965147) as well SM 281/30-1 and SM 281/32-1. Furthermore, the authors acknowledge the support of the TUM Innovation Network "CoConstruct", the DFG Transregio 277 "Additive Manufacturing for Construction" (grant number 414265976), and the EU Pathfinder project "AM2PM".

References

[1] J. Stührenberg and K. Smarsly. LIO-BIM – coupling LiDAR inertial odometry with building information modeling for robot localization and mapping. *Advanced Engineering Informatics*, 66:103477, 2025.

[2] K. Smarsly and K. Dragos. Advancing civil infrastructure assessment through robotic fleets. *Internet of Things and Cyber-Physical Systems*, 4:138–140, 2024.

[3] A. Borrmann, T. Bruckmann, K. Dörfler, T. Hartmann, and K. Smarsly. Towards realizing the information backbone of robotized construction–computational methods and cyber-physical architectures for collaborative robotic fleets. In *Proc. of the CIB W78 Conf. 2021*, 2021.

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

[5] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng. Ros: An open-source robot operating system. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Software*, 2009.

[6] A. Ibrahim, A. Sabet, and M. Golparvar-Fard. BIM-driven mission planning and navigation for automatic indoor construction progress detection using robotic ground platform. In *Proc. of the 2019 European Conf. on Computing in Construction*, pages 182–189, 2019.

[7] B.R. Mantha, M.K. Jung, B.G. de Soto, C.C. Carol, and V.R. Kamat. Generalized task allocation and route planning for robots with multiple depots in indoor building environments. *Automation in Construction*, 119, 2020.

[8] R. Rust, G. Casas, S. Parascho, D. Jenny, K. Dörfler, M. Helmreich, A. Gandia, Z. Ma, I. Ariza, M. Pacher, B. Lytle, Y. Huang, C. Kasirer, E. Bruun, and P.Y.V. Leung. COMPAS FAB: Robotic fabrication package for the compas framework. https://github.com/compas-dev/compas_fab/, 2018. Gramazio Kohler Research, ETH Zürich.

[9] J.G. Everett and A.H. Slocum. Automation and robotics opportunities: Construction versus manufacturing. *Journal of Construction Engineering and Management*, 120, 1994.

[10] K.M. Lundeen, V.R. Kamat, C.C. Menassa, and W. McGee. Autonomous motion planning and task execution in geometrically adaptive robotized construction work. *Automation in Construction*, 100, 2019.

[11] L. Ding, W. Jiang, C. Zhou, and S. Liu. BIM-based task-level planning for robotic brick assembly through image-based 3d modeling. *Advanced Engineering Informatics*, 43, 2020.

[12] S. Kim, M. Peavy, P. Huang, and K. Kim. Development of BIM-integrated construction robot task planning and simulation system. *Automation in Construction*, 127, 2021.

[13] A. Zhu, P. Pauwels, and B. De Vries. Smart component-oriented method of construction robot coordination for prefabricated housing. *Automation in Construction*, 129: 103778, 2021.

[14] A. Zhu, P. Pauwels, and B. De Vries. Component-based robot prefabricated construction simulation using ifc-based building information models. *Automation in construction*, 152:104899, 2023.

[15] A. Zhu, T. Dai, G. Xu, P. Pauwels, B. De Vries, and M. Fang. Deep reinforcement learning for real-time assembly planning in robot-based prefabricated construction. *IEEE Transactions on Automation Science and Engineering*, 20(3):1515–1526, 2023.

[16] T. Wrabel, S. Esser, and A. Borrmann. A framework architecture for the information backbone for robotic construction. In *Proc. of the 43rd Int. Symp. on Automation and Robotics in Construction (ISARC 2026)*, 2026.

[17] M. Brinkhoff, S. Esser, and A. Borrmann. An integrated product–process representation for deriving robotic task structures using graph transformations. In *Proc. of the 43rd Int. Symp. on Automation and Robotics in Construction (ISARC 2026)*, 2026.

[18] M. Slepicka, S. Vilgertshofer, and A. Borrmann. Fabrication information modeling: Interfacing building information modeling with digital fabrication. *Construction Robotics*, 6(2):87–99, 2022.

[19] I-C. Wu, A. Borrmann, U. Beissert, M. König, and E. Rank. Bridge construction schedule generation with pattern-based construction models and constraint-based simulation. *Advanced Engineering Informatics*, 24(4):379–388, 2010.

[20] T. Bruckmann and R. Boumann. Simulation and optimization of automated masonry construction using cable robots. *Advanced Engineering Informatics*, 50:101388, 2021.

[21] U. Beißert, M. König, and H.-J. Bargstädt. Constraint-based simulation of outfitting processes in building engineering. In *Proc. of the 24th CIB-W78 Conf.*, 2007.

[22] R. Zlot and A. Stentz. Market-based multirobot coordination for complex tasks. *The Int. Journal of Robotics Research*, 25(1):73–101, 2006.

[23] M. Colledanchise and P. Ögren. *Behavior Trees in Robotics and AI: An Introduction*. CRC Press, Clermont, Florida, 2018.

[24] A. Tandon, J. Stührenberg, and K. Smarsly. Automated building inspections coupling building information modeling and behavior trees. In *Proc. of the 2025 European Conf. on Computing in Construction*, 2025.

[25] M.R. Kolani, A. Mirhaghgoo, S. Nousias, and A. Borrmann. Model-based adaptive controller for fast and restricted swing control in tower cranes. In *Proc. of the 43rd Int. Symp. on Automation and Robotics in Construction (ISARC 2026)*, 2026.