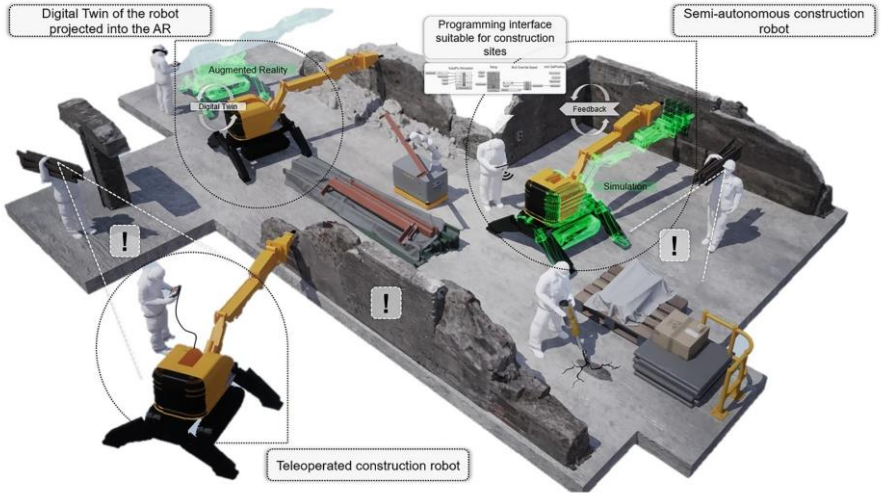
Development of an intuitive 3D Human-Machine-Interface for Construction-Site Robotic Task Planning

Software Development and Human-Robot Interaction

19.05.2025



Brell-Cokcan, S., Lee, H.J. (2022). Robotics in Construction. In: Ang, M.H., Khatib, O., Siciliano, B. (eds) Encyclopedia of Robotics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41610-1_218-1

Background

Modern construction sites will increasingly rely on semi-autonomous robots to perform repetitive or hazardous tasks such as drilling, brick-laying, or material transport. While the underlying hardware (robot arms, mobile bases, depth cameras, LiDAR) has matured, intuitive human-machine interfaces (HMIs) that let site engineers plan, preview, and certify robot actions are still missing. Web-based 3D technologies—especially **Three.js** combined with **TypeScript**—offer a lightweight, cross-platform path toward digital-twin visualization and interaction without requiring specialized desktop software. An HMI that clearly shows the robot, its sensed environment, and planned trajectory before execution can greatly improve safety, reduce downtime, and streamline certification workflows.

Description

This thesis investigates interaction modalities and implements a browser-based 3D interface that allows users to:

- load scans of the construction environment (point clouds, meshes),

SUPERVISOR

Prof. Dr.-Ing. Matthias Althoff

ADVISOR

M.Sc. Julian Hoffmann

TYPE

Bachelor's thesis

LANGUAGE

English

CONTACT

julian.a.hoffmann@tum.de

REQUIREMENTS

Programming: TypeScript / JavaScript, Three.js, Node.js
Robotics Frameworks: ROS 2 (Jazzy), URDF/XACRO basics Tools
Libraries: WebGL, glTF, PCD/PLY handling, WebSockets (rosbridge), Git
Optional: UX/UI design fundamentals

- place and configure construction-site robots (mobile manipulator(s) or arm on rail),
- define or import high-level tasks (e.g., "drill holes along beam"),
- visualize the resulting motion plan (trajectory, speed, tool orientation), and
- preview the sequence in real time to obtain certification sign-off *before* physical execution.

The core delivers a **TypeScript + Three.js** application communicating with ROS 2 (Jazzy) back-end services for robot state, planning, and execution.

Tasks

- **Literature & Technology Review**: HMI design for robotics, digital-twin standards, Three.js best practices, construction-site safety norms (e.g., ISO 10218, ISO 21939).
- **System Architecture**: Define data flow between ROS 2 nodes, WebSocket/ROSbridge, and the Three.js front-end. Specify environment and robot model formats (URDF, glTF, PCD, etc.).
- **Interface Implementation**: – Environment renderer with dynamic level-of-detail point-cloud streaming. – Robot visualisation with real-time joint state updates and configurable end-effector tools. – Task authoring tools (way-point editor, drag-&-drop primitives). – Trajectory preview (animated path, speed/acceleration heat-map, collision overlays).
- **Input-Modality Research & Usability Study**: Prototype alternative controls; conduct a small user study (≥ 6 participants) measuring task-completion time, error rate, and System Usability Scale (SUS).
- **Documentation & Presentation**: Detailed code documentation, user manual, thesis report, and defense.

Deliverables

1. **Web-based 3D HMI** (source code, build scripts).
2. **Integration package** for ROS 2 Jazzy (message definitions, launch files).
3. **Evaluation report** containing usability-study results and certification-workflow demonstration.
4. **Bachelor thesis document** covering background, design, implementation, and evaluation.

Further Readings

- Czarnowski, Jan, et al. "Technology gaps in human-machine interfaces for autonomous construction robots." *Automation in Construction* 94 (2018): 179-190.
- Threejs library: https://threejs.org/