

Re-Using Design-Time Artifacts for Model-Based Digital Twins in Collaborative Assembly

Jeshwitha Jesus Raja, Shaza Elbishbishy, Yanire Gutierrez, Philipp Kranz and Marian Daun

Center for Robotics, Technical University of Applied Sciences Würzburg-Schweinfurt, Schweinfurt, Germany

Abstract

Collaborative robots are increasingly adopted in manufacturing to improve flexibility and productivity, but seamless human-robot collaboration requires mechanisms that prevent errors and ensure safety. Unlike conventional 3D-based digital twins that simulate the physical system, a model-based digital twin captures the logical flow of tasks, dependencies, and interactions, offering a lightweight and scalable solution for adaptive monitoring and control. This paper presents a model-based digital twin that reuses design-time artifacts, namely Goal Models and Process Models, for runtime monitoring, task synchronization, and error detection. The framework is evaluated on a collaborative assembly process.

Keywords

Human-Robot Collaboration, Digital Twin, Goal Model, Process Model

1. Introduction

Human-Robot Collaboration (HRC) is an emerging paradigm in modern manufacturing that enables humans and collaborative robots (cobots) to work side by side, offering flexibility and dynamic task sharing [1]. However, ensuring safety and preventing errors in such workspaces remain a key challenge, requiring additional monitoring and coordination mechanisms [2].

Digital twins address this challenge by providing a virtual representation of the physical system that supports runtime insight and dynamic decision-making. Unlike conventional 3D-based digital twins, which capture the physical aspects of the system, model-based digital twins focus on the logical flow of tasks, dependencies, and interactions, and can thereby enable precise task execution, early error detection, and continuous system improvement [3, 4]. Building on this, we propose the reuse of design-time artifacts for the development of model-based digital twins. Goal-Oriented Requirements Language (GRL) is used to capture system objectives, dependencies, and constraints. Business Process Model and Notation (BPMN) is used to specify task sequences, flows, and interactions. We evaluate the approach on an HRC use case.

2. Related Work

HRC combines the complementary strengths of robots and humans, offering increased productivity and reduced costs [5], provided that safe and reliable interaction can be ensured. Kritzing et al. [6] showed the broad use of digital twins in industry but pointed out that many implementations operate only at the digital shadow level, with limited reciprocal data exchange. In HRC specifically, ensuring dependable runtime synchronization between physical and virtual systems is still a significant challenge, largely because accurately modeling a cobot is difficult in a human-involved environment with inherent uncertainty. This also limits scalability in highly dynamic scenarios [7].

Model-Based Engineering, the structured use of models, methods, and tools across a system's lifecycle, offers a promising direction here and can be applied effectively to the creation and use of digital twins [8].

ER2026: Companion Proceedings of the 45th International Conference on Conceptual Modeling: Posters and Demos, October 05-08, 2026, St. John's, NL, Canada

✉ jeshwitha.jesusraja@study.thws.de (J. Jesus Raja); marian.daun@thws.de (M. Daun)

ORCID 0009-0008-7886-7081 (J. Jesus Raja); 0009-0002-0975-272X (S. Elbishbishy); 0009-0007-2959-7241 (Y. Gutierrez); 0000-0002-1057-4273 (P. Kranz); 0000-0002-9156-9731 (M. Daun)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

This paper builds on this by using GRL goal models and BPMN process models, both established in requirements engineering and previously applied to robotic and collaborative systems [9, 10], as the artifacts underlying the digital twin.

3. Model-Based Digital Twin

A model-based digital twin reuses design models to monitor and control collaborative assembly processes at runtime, as illustrated in Figure 1. At runtime, the physical system, comprising the human, cobot, and workspace, interacts with the digital twin, which consists of a goal model and a process model. Runtime data is continuously exchanged between the physical system and digital twin, enabling process monitoring and fault detection. The goal model represents the tasks performed by the human and cobot, while the BPMN process model captures their order, decision points, and interaction points, making process deviations visible. Robot sensor data (e.g., joint positions) and human signals are continuously mapped to both models, keeping objectives and workflows synchronized with physical execution and enabling model state changes to trigger robot commands [4].

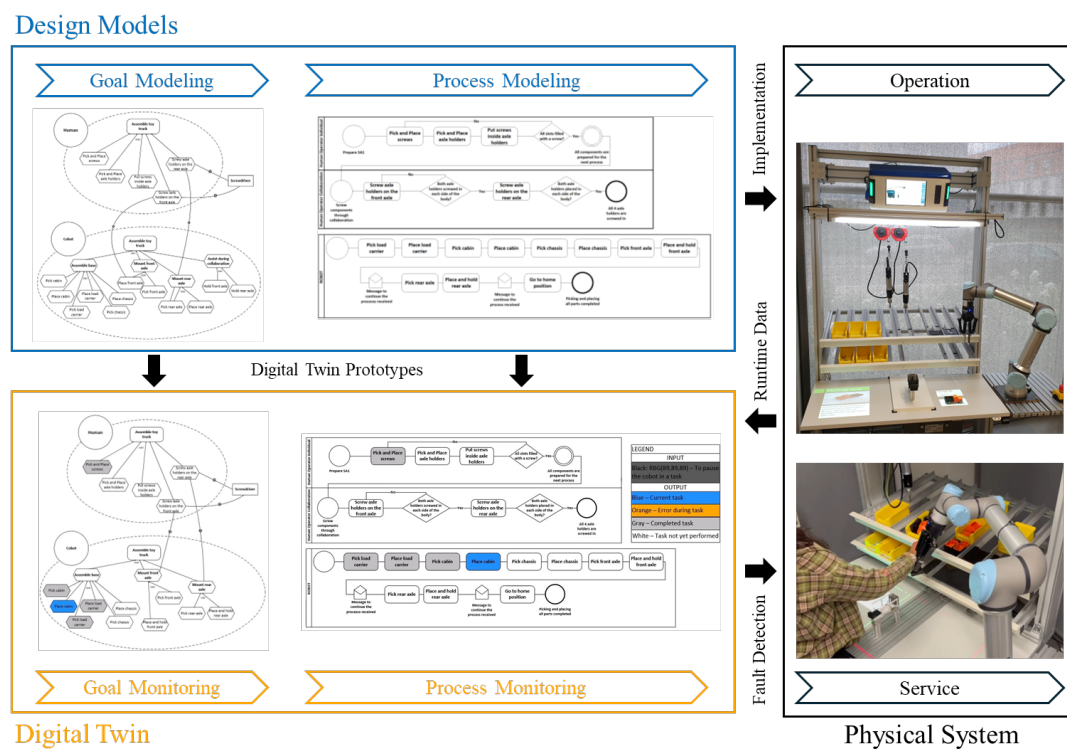


Figure 1: Concept of the model-based digital twin.

4. Implementation and Evaluation

4.1. Use Case

For evaluating the framework, a toy truck assembly process is considered. The process starts with the cobot picking and placing truck components. The human prepares four axle holders and then signals completion through a virtual button. The cobot then places the front axle on the chassis, aligned with

the screw entries, and holds it in position while the human mounts the axle holders on both sides and screws them into place. The same is then done with the rear axle.

4.2. Design-Time Models

Figure 2 shows the goal model, which represents system tasks, dependencies, and expected outcomes as goals. “Assemble Toy Truck” is the main goal for both the human and the cobot, but the tasks required to fulfill it differ between them. The order of execution is visualized with the dependency element. Figure 3 shows the process model, specifying the precise order of activities, decision points, and human-robot interactions. Tasks specific to the human and the cobot are presented in separate lanes, with an additional lane for the collaborative processes. Task completions signaled between the human and the cobot are represented via *message* elements.

The task highlighting scheme uses color codes to differentiate task status: white for a task that has not been started, blue for a task in progress and running without problems, gray for a completed task, orange for interruptions and errors such as a collision between the human and the cobot, and black for halting points at which the human worker wants the system to pause execution.

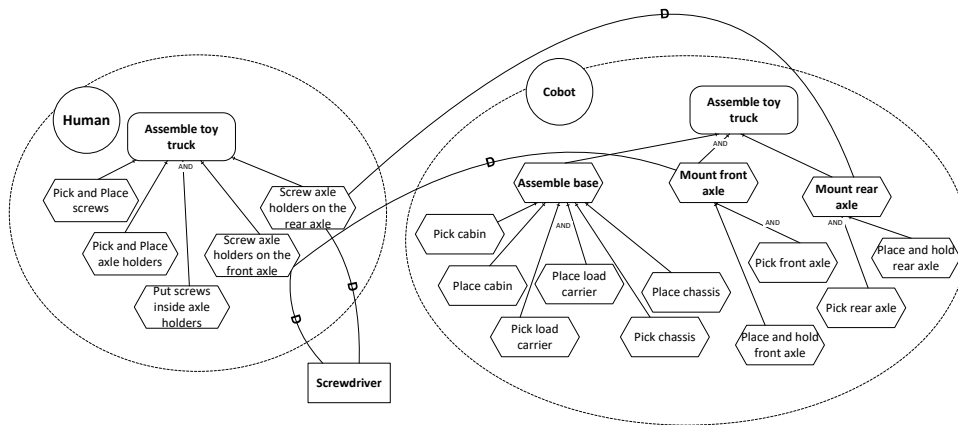


Figure 2: Goal model of the collaborative assembly process.

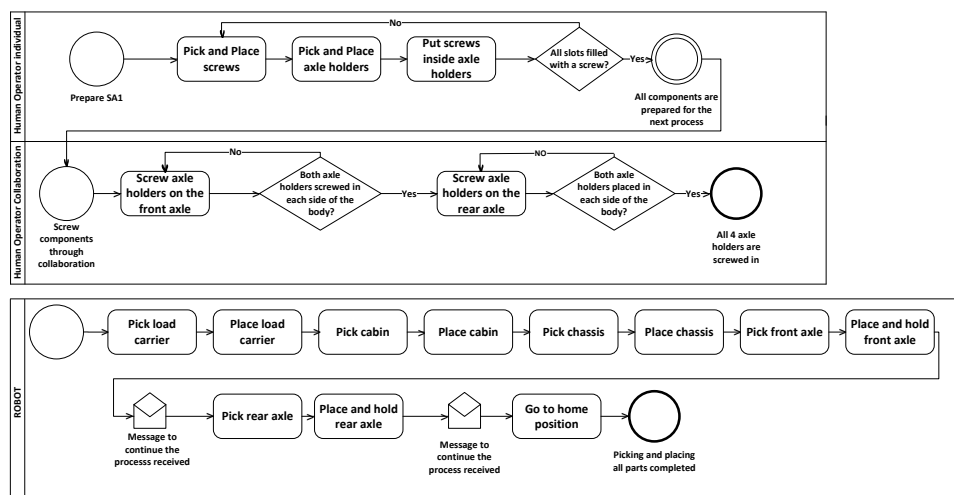


Figure 3: Process model of the collaborative assembly process.

4.3. Technical Setup

The cobot is monitored via the RTDE (Real-Time Data Exchange) library, which transfers joint positions, velocities, and task progress. Human task completion is captured and transmitted over the same interface. A Python component links RTDE, the models, and the cobot's dashboard server, cleaning and structuring the data before it reaches Microsoft Visio, which serves as the model editor for runtime visualization.

4.4. Runtime Evaluation and Findings

The evaluation examines two aspects: monitoring (runtime feedback on task progress, goal achievement, and safety) and control (human-initiated pause, resume, or adjustment through model-based interactions). For monitoring, sub-tasks, tasks, and goals in the goal model and activities in the process model are highlighted during runtime. If a collision occurs, whether between the human and the cobot or with the environment, the cobot performs a safety stop and the affected task is highlighted in orange. For control, the human changes a task shape to black in either model. This sends a control signal to the cobot, instructing it to pause at the corresponding task. The cobot remains at halt until further input is received. The structured models provided visibility of task progress for both the human and the cobot, dynamic highlighting made task, error states and their location in the process immediately apparent, and pausing and resuming tasks allowed controlled interventions without disrupting the overall workflow.

5. Conclusion and Future Work

In this paper we presented a model-based digital twin framework for collaborative assembly that reuses design-time artifacts, namely GRL goal models and BPMN process models. Integrating robot sensor data and human signals with these models enables synchronized monitoring, task highlighting, and adaptive control of HRC, and makes it plausible to influence robot behavior by changing task states. For future work, the approach needs to be applied to more complex manufacturing processes and additional robotic tasks.

Declaration on Generative AI

No generative AI tools were used in the preparation of this manuscript.

References

- [1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, O. Khatib, Progress and prospects of the human-robot collaboration, *Autonomous robots* 42 (2018) 957–975.
- [2] N. Kozamernik, J. Zaletelj, A. Košir, F. Šuligoj, D. Bračun, Visual quality and safety monitoring system for human-robot cooperation, *The International Journal of Advanced Manufacturing Technology* 128 (2023) 685–701.
- [3] J. Jesus Raja, S. Elbishbishy, Y. Gutierrez, I. Mohamed, P. Kranz, M. Daun, Model-based digital twin for collaborative robots, in: *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics*, SCITEPRESS-Science and Technology Publications, 2024, pp. 256–263.
- [4] F. Tao, B. Xiao, Q. Qi, J. Cheng, P. Ji, Digital twin modeling, *Journal of Manufacturing Systems* 64 (2022) 372–389.
- [5] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, G. Rosati, Human-robot collaboration in manufacturing applications: a review, *Robotics* 8 (2019) 100.
- [6] W. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital twin in manufacturing: A categorical literature review and classification, *Ifac-PapersOnline* 51 (2018) 1016–1022.

- [7] A. K. Ramasubramanian, R. Mathew, M. Kelly, V. Hargaden, N. Papakostas, Digital twin for human–robot collaboration in manufacturing: Review and outlook, *Applied Sciences* 12 (2022) 4811.
- [8] J. Jesus Raja, M. Manjunath, M. Daun, Towards a goal-oriented approach for engineering digital twins of robotic systems., in: *ENASE*, 2024, pp. 466–473.
- [9] S. Elbishbishy, J. Jesus Raja, P. Kranz, M. Daun, Combining goal and process models for the specification of human-robot collaborations, in: *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering*, SCITEPRESS-Science and Technology Publications, 2025, pp. 217–228.
- [10] F. Corradini, S. Pettinari, B. Re, L. Rossi, F. Tiezzi, A bpmn-driven framework for multi-robot system development, *Robotics and Autonomous Systems* 160 (2023) 104322.