Model-Based Digital Twin for Collaborative Robots

Jeshwitha Jesus Raja[a], Shaza Elbishbishy[b], Yanire Gutierrez[c], Ibrahim Mohamed, Philipp Kranz[d] and Marian Daun[e]
Center of Robotics, Technical University of Applied Sciences Würzburg-Schweinfurt, Schweinfurt, Germany

Keywords: Collaborative Robot, Digital Twin, Model-Based Engineering, Conceptual Modeling.

Abstract: Industry 4.0 is reshaping the way individuals live and work, with significant impact on manufacturing processes. Collaborative robots, or cobots-designed to be easily programmable and capable of directly interacting with humans-are expected to play a critical role in future manufacturing scenarios by reducing setup times, labor costs, material waste, and processing durations. However, ensuring safety remains a major challenge in their industrial application. One promising solution for addressing safety, as well as other real-time monitoring needs, is the digital twin. While the potential of digital twins is widely acknowledged in both academia and industry, current implementations often face several challenges, including high development costs and the lack of a systematic approach to ensure consistency between the physical and virtual representations. These limitations hinder the widespread adoption and scalability of digital twins in industrial processes. In this paper, we propose a model-based approach to digital twins, which emphasizes the reuse of design-time models at runtime. This ensures a coherent relationship between the physical system and its digital counterpart, aiming to overcome current barriers and facilitate a more seamless integration into industrial environments.

1 INTRODUCTION

While there is still no consensus on a concrete definition for a digital twin (Zhang et al., 2021), it is commonly understood that a digital twin is a virtual representation of a physical object or system that accurately reflects its real-world counterpart. Initially defined by IBM to encompass an object's entire life cycle, a digital twin is continuously updated with real-time data. The applications of digital twins include real-time monitoring, design and planning, optimization, maintenance, and remote access. The use of digital twin technology is expected to grow exponentially in the coming decades (Tao et al., 2022). However, the current literature on digital twins primarily focuses on technical approaches, analysis methodologies, and the challenges associated with data collection and integration into digital twins (VanDerHorn and Mahadevan, 2021). Despite these advancements, there is a notable lack of structured approaches for creating digital twins and visual representations that foster better understanding and usability of a system (cf. (Sandkuhl and Stirna, 2020)).

In this paper, we address this challenge by proposing a concept for a model-based digital twin. This approach allows us to reuse design time artifacts and highlight system execution, predicted defects, and other analysis results within the existing artifacts. This enables runtime inspection of the digital twin—and thereby the system monitored—and supports iterative development and continuous improvement of the robotic application. Reusing conceptual models from the design phase also has the advantage as descriptive tools for highlighting data input are already in existence (Cañas et al., 2021).

Therefore, in this paper, we introduce our concept for a model-based digital twin for a collaborative robot used in a manufacturing scenario. By using standard interfaces of the robot and other monitoring systems and connecting them with current standard model editors, we can reuse design models for runtime monitoring and analysis. This contributes to the enhancement of system development effectiveness through a model-based digital twin, which facilitates continuous system improvement and reduces development costs by iteratively reusing design artifacts.

This paper is structured as follows: In Section 2,

[a] https://orcid.org/0009-0008-7886-7081
[b] https://orcid.org/0009-0002-0975-272X
[c] https://orcid.org/0009-0007-2959-7241
[d] https://orcid.org/0000-0002-1057-4273
[e] https://orcid.org/0000-0002-9156-9731

Raja, J., Elbishbishy, S., Gutierrez, Y., Mohamed, I., Kranz, P. and Daun, M.
Model-Based Digital Twin for Collaborative Robots.
DOI: 10.5220/0013016400003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 256-263
ISBN: 978-989-758-717-7; ISSN: 2184-2809

research related to the concept of digital twin architecture and conceptual modeling, as well as its implementation in industry automation, is discussed. Section 3 introduces the use case to be used throughout the remainder of the paper. Section 4 introduces the approach. Finally, Section 5 concludes the paper along with future work.

2 RELATED WORK

Digital twins replicate real systems in a virtual space to improve performance through advanced analysis and prediction techniques (Zhang et al., 2021). Driven by data and models, digital twins can perform tasks like monitoring, simulation, prediction, and optimization (Tao et al., 2022). Specifically, digital twin modeling is crucial for accurately portraying the physical entity, enabling the digital twin to deliver functional services and meet application requirements. Challenges include creating reliable models, ensuring real-time communication, and developing deep analysis methods (VanDerHorn and Mahadevan, 2021).

However, as Kritzinger et al. (Kritzinger et al., 2018) noted, the development of digital twins is still in its initial stages, with literature mainly presenting conceptual ideas without concrete case studies. While digital twin technology has advanced since Kritzinger's investigation in 2018, it still requires more structure and planning to achieve its full potential (Ramasubramanian et al., 2022). Although there are many papers on digital twins for manufacturing systems, there is still insufficient definitive evidence of digital twin implementation and evaluation in the industry (Redelinghuys et al., 2020). One reason for this is the lack of systematic approaches to developing digital twins (Sandkuhl and Stirna, 2020).

Current approaches like AutomationML(Drath, 2021) and Asset Administration Shell (AAS)(Tantik and Anderl, 2017) offer frameworks for integrating various aspects of industrial automation systems. AutomationML, for instance, facilitates the exchange of plant engineering information, while AAS serves as a digital representation of an asset's lifecycle. Our work differentiates itself from the above-mentioned approaches by focusing on a model-based digital twin specifically designed for collaborative robots in manufacturing scenarios.

AutomationML primarily addresses data exchange rather than real-time monitoring and dynamic interaction between physical and virtual entities. In contrast, our model-based digital twin approach not only ensures data consistency and integration but also supports real-time updates and interaction, which are critical for the continuous improvement of robotic applications. Similarly, the Asset Administration Shell (AAS) framework, as part of the Industry 4.0 initiative, provides a robust structure for asset management as it often requires extensive customization to suit specific applications and lacks inherent support for runtime monitoring and predictive analysis tailored to collaborative robots.

In the industrial domain, there already are model-based artifacts from the specification and development of a system. Models can contain the potential to depict a system's functions, structure, or even its behavior (Davis, 1993). Some existing modeling languages that have proven useful for modeling robotic systems are goal models (Mussbacher and Nuttall, 2014) and process models (Petrasch and Hentschke, 2016). The Goal-Oriented Requirements Language (GRL(ITU Int. Telecommunication Union, 2018)) can be used to elaborate and document the requirements of a physical system in the early stages of development (Daun et al., 2019; Daun et al., 2021). Business Process Model and Notation (BPMN), on the other hand, can play an important role in documenting the operations and requirements of a process during its operation in industry. Both modeling languages also contain potential with respect to safety analysis (Khan et al., 2015).

In previous work, we have successfully extended the GRL to support early safety assessment of human robot collaborations (Daun et al., 2023; Manjunath et al., 2024). As a digital twin is also seen as an important safety mechanism in the field, we also already proposed the use of goal models to systematically develop digital twins (cf. (Jesus Raja et al., 2024)). In this paper, we propose to go one step further, by using the design time models as runtime models monitored by the digital twin.

3 RUNNING EXAMPLE

Our case example explores the collaborative dynamics between human operators and cobots in assembly processes. The assembly process includes operations ranging from picking and preparing components to screwing and placing them. In the exemplary production process, a toy truck is produced through the collaboration between a human and a robot. The assembly process is as follows:

1. **Cabin (C2) and load carrier (C1) are lying upside down in the mounting bracket:** Cobot begins by carefully picking the load carrier from the storage area and positioning it upside down in

the collaborative workspace. It then picks up the cabin and places it in the workspace next to the load carrier with precision.

2. **Chassis (C3) is placed on top:** After placing the structure of the truck (C1 and C2) on the collaborative area, the cobot picks up the chassis and places it carefully with precision on the truck structure.
3. **Screws (C7) are placed inside the axle holders (C6):** While the cobot manages the initial tasks, the human operator prepares the parts for screwing by inserting two screws into each axle holder, repeating this process for all four holders. The finished product of this process is called subassembly 1. Once the human operator completes the task, they signal the computer that the work is finished by pressing a virtual button.
4. **Front axle (C4) and rear axle (C5) are aligned on top of the chassis:** The cobot retrieves the front axle and places it on the chassis, aligning with the screw's entries. The cobot securely holds the front axle in place and waits for the human operator's intervention. After the completion of subassembly 1, the operator moves to the collaborative area and with the use of the screwdriver then screws the axle holders, starting with the front right side, followed by the front left side. After this process is completed, the human operator signals the computer through the virtual button that the work is finished and so, the same process is again repeated for the rear axle. First, the cobot picks up the rear axle and then secures it in place on top of the chassis, aligning it with the holes for the screws. After that, the human enters the collaborative area to screw the axle holder, starting from the right side.

4 MODEL-BASED DIGITAL TWIN

4.1 Problem Statement

In industrial environments, collaborative robots are increasingly deployed to enhance productivity and adaptability. However, current monitoring systems predominantly rely on basic status updates and data logging, which provides limited visibility into the cobots' real-time operations and interactions with human operators (Das et al., 2009). These traditional systems often lack to offer an in-depth, dynamic representation of ongoing tasks, making it challenging to optimize workflows, predict issues, and ensure seamless collaboration (Bruno and Antonelli, 2018).

4.2 Solution Idea

To address these limitations, we propose the introduction of a model-based digital twin for collaborative robots, represented through detailed conceptual models that dynamically illustrate the entire process and highlight the current tasks. This approach leverages real-time data obtained via the standard robot interface provided by the robot manufacturer. The digital twin utilizes models created using a standard model editor. Currently, we are developing the digital twin mainly for monitoring and real-time analysis, without making changes to the physical twin. "Digital shadow" is a more precise term for this one-way flow of information (Bergs et al., 2021). However, for the future we plan to develop the digital shadow into a real digital twin with advanced analysis and the ability for interventions and adaptations in the physical twin.

The proposed digital twin will employ BPMN and GRL goal models to represent different aspects of the workflow. BPMN models will be used to map out the entire process flow, detailing each step and decision point. The reason we choose BPMN is that it provides a standardized, visual representation of the workflow, detailing each step of the cobot as it picks, places and holds components. This clarity facilitates better communication and process improvement.

Figure 1 shows the basic GRL goal model that showcases the goals, actors, tasks and dependencies of the use case introduced in Section 3. The process for the execution of tasks can be seen in the BPMN process model shown in Figure 2. The two actors are the human and the robot, and their tasks are divided based on the parts they handle. This process involves collaborative human-robot interaction, which can be visualized through dependencies in the GRL goal model.

Goal models will be employed to focus on specific objectives and their achievement. Goal models focus on specific objectives. The advantage of using these models lies in their ability to enable real-time progress tracking and decision support, allowing dynamic adjustments based on the current state of assembly.

4.3 Architecture

The current concept for creating the model-based digital twin can be seen in Figure 3 and is described in detail in the following subsection.

The architecture of the physical twin is shown in Figure 4. It begins with the complete system, the 'Spatial Augmented Reality System', which is later

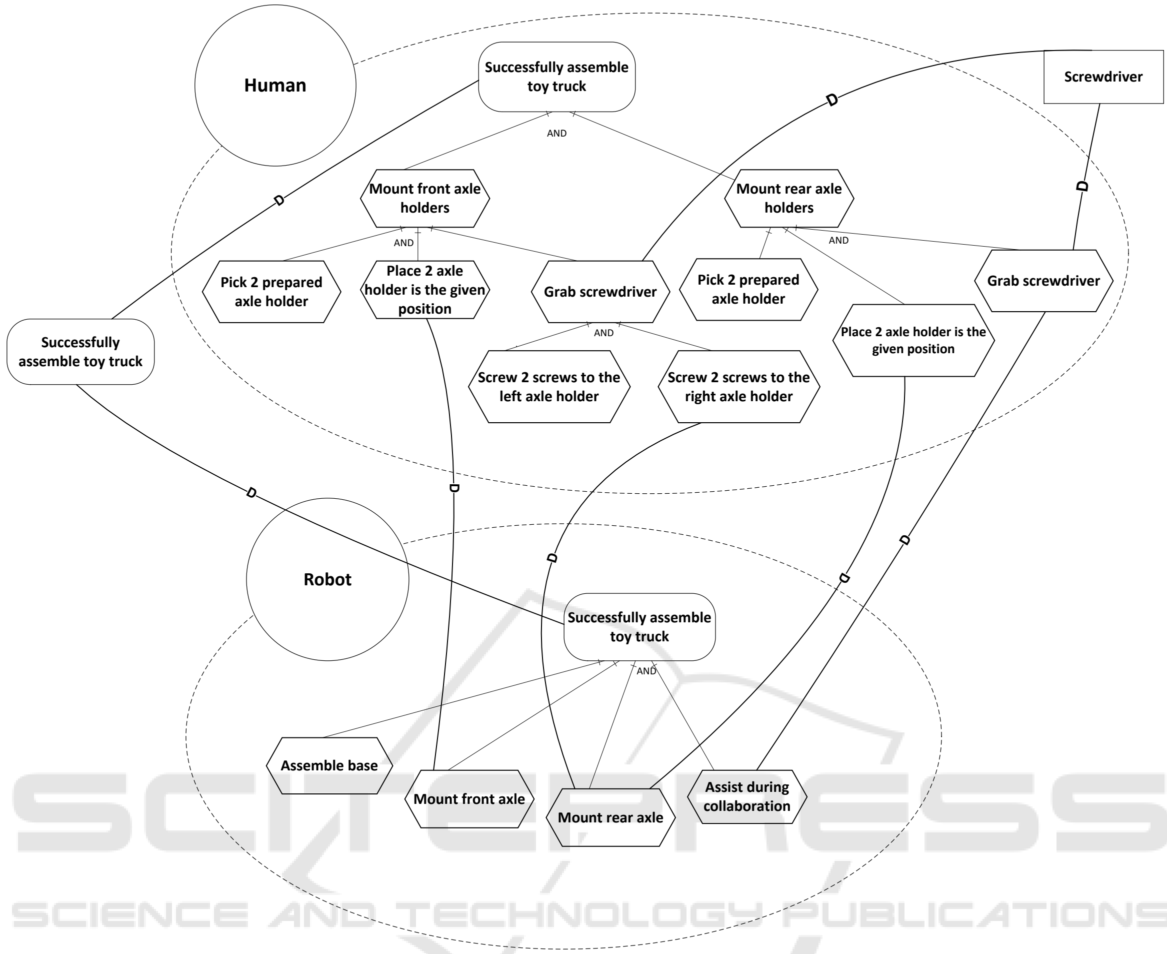

Figure 1: Goal Model of the Truck Assembly use case.

divided into four components: ‘Screwdriver’, ‘Human Operator’, ‘Monitoring System’, and ‘Cobot’. The monitoring system consists of a projector and a camera. This projector contains a depth sensor, and together with the human operator, it facilitates Human-Computer communication. The cobot also includes a control tablet and a torque sensor. The control tablet, along with the monitoring system, facilitates Robot-Computer communication. This illustrates that there is no direct communication between the human and the robot but rather through the monitoring system.

4.4 Collection of Input and Output Data

Analyzing and understanding the architecture provides information on the data needed to build the digital twin. Therefore, the second step (shown in Figure 3) is to collect output data from the cobot and the human operator for runtime analysis and monitoring. In our case, we use a Universal Robot with an RTDE (Real Time Data Exchange) interface to control and receive data from the cobot. Data from human actions is collected from a depth sensor integrated with a virtual green button that the human clicks after each task. This data, along with RTDE information, must be processed through the interface and provided to the models. For now, we only use the information from the robot for processing and visualizing the models.

4.5 Data Pre-Processing

The data collected from cobot and human operator can be exported and processed for use in the models through Python, which is shown as the third step in Figure 3. A Python adapter software serves as the interface between the physical twin and the model-based digital twin. The RTDE interface that was used to collect the needed data from the cobot can be utilized with Python through the provided Python bind-

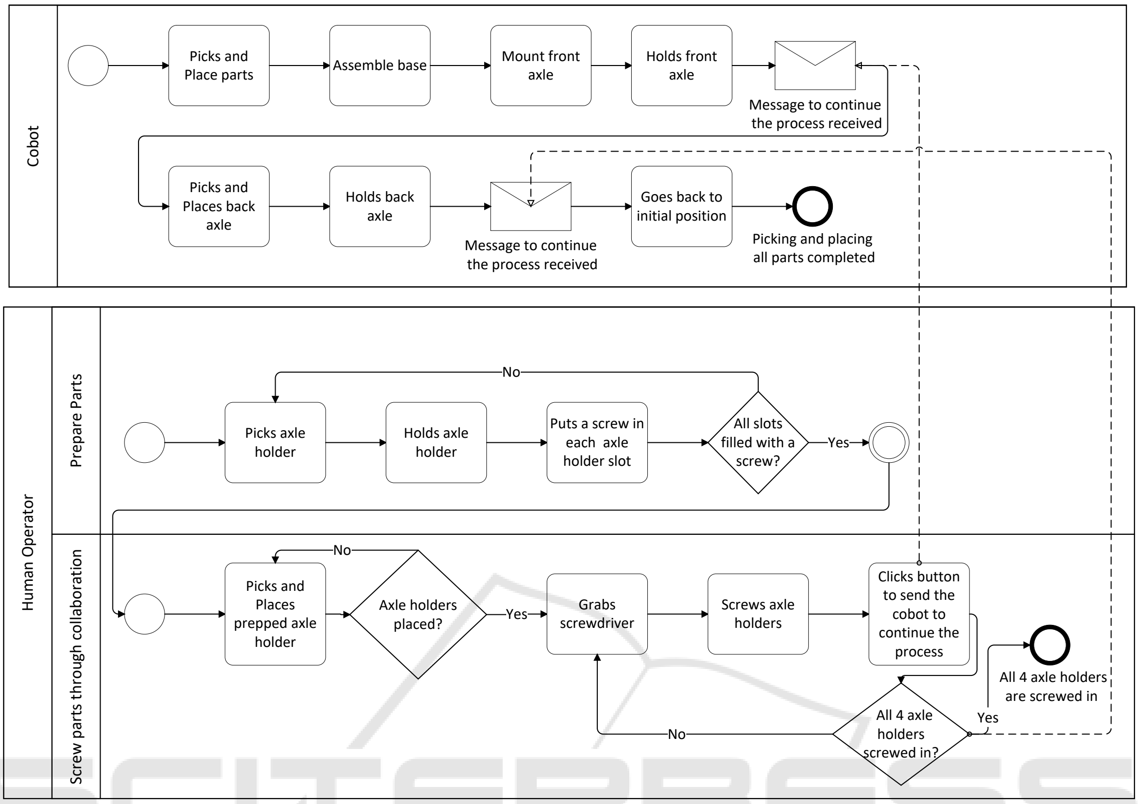

Figure 2: Business Process Model of the Truck Assembly use case.

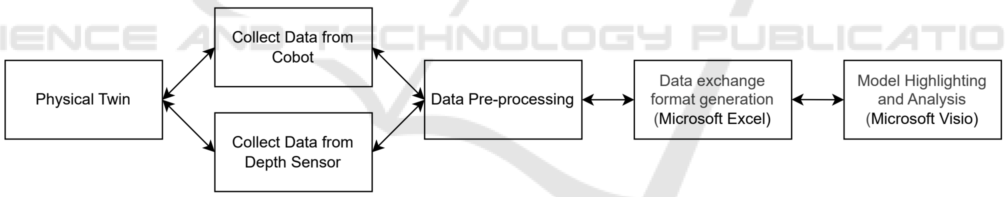

Figure 3: Concept for developing a model-based digital twin.

ings. The required data from the cobot can be individually collected using predefined functions and then compiled. Some of the data variables that can be collected and used include:

- *timestamp*
- *target_q (target position)*
- *actual_q (current position)*
- *actual_digital_input_bits*

Next, the data is exported into a format to be reused as standard input by the chosen standard model editor. In our case, we chose to use *Microsoft Visio* as Model Editor, as it is commonly available and accessible in academia and industry.

Therefore, we generate a *Microsoft Excel* file from the processed RTDE data, which is then automatically imported into Microsoft Visio. Using Python and the RTDE interface, we collect the specified variable, and with this data, we create two columns in the Excel document. The first column corresponds to the timestamp, and the second column corresponds to the current task. The current task is labeled based on the current position, and differentiation between human and robot tasks is achieved using digital input bits. After extracting the data into the Excel document, it can be used in Microsoft Visio.

4.6 Model Highlighting

The representation of the models (GRL goal model and BPMN) is shown in Microsoft Visio. Model highlighting to show runtime representations can be

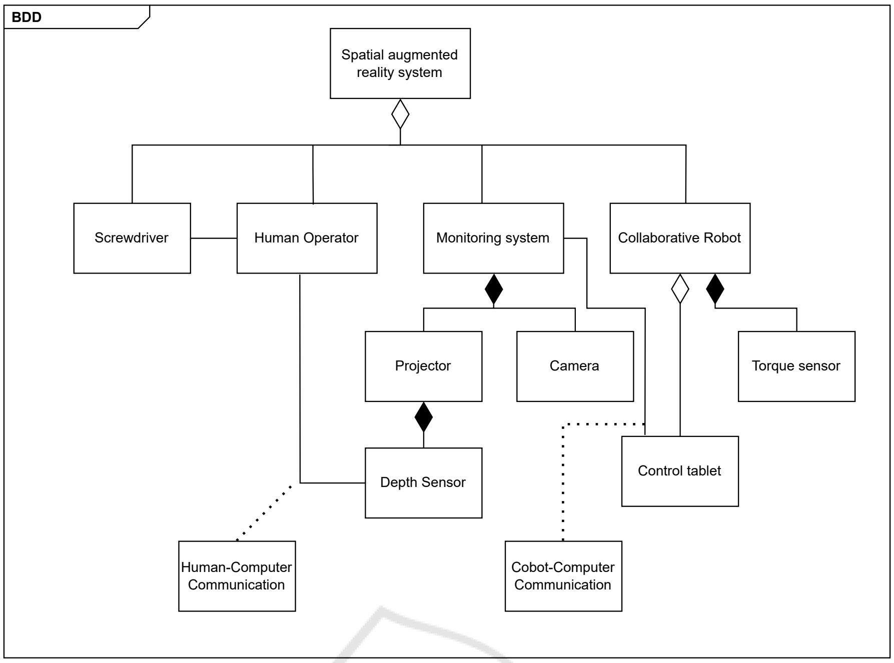

Figure 4: The architecture of the physical system of the collaborative workspace.

achieved directly in Microsoft Visio using the 'Link data to shapes' function. These shapes are assigned conditions, and based on those conditions, the currently implementing task or process can be highlighted. This means that the model created fulfills the properties of a digital shadow. In order to develop this into a digital twin in our future work, more advanced analyses can also be directly implemented in Microsoft Visio using C#-implemented add-ons.

Figure 5 shows how tasks can be highlighted in the goal model. Here, bidirectional dependencies indicate tasks where both actors are involved and mutually dependent on each other. For instance, tasks such as the human performing 'Screw 2 screws to the right axle holder' and the robot holding the 'rear axle' are collaborative and connected by bidirectional dependencies.

Tasks highlighted in green indicate that they are currently being executed. Other colors can be used to show different statuses of the tasks, such as 'completed,' 'waiting,' and 'needs attention.' For example, the task 'place one axle holder on the right of the front axle' is currently being executed and is highlighted green. The colors and semantics can be customized based on the type of process execution and monitoring.

To achieve this, we utilize Visio's 'Link Data to Shapes' feature. After uploading the Excel document to the Visio file, this feature allows us to link a cell to either a main task or a sub-task. By utilizing this capability, we ensure that the displayed statuses accurately reflect the current state of each task.

Furthermore, the use of colors such as blue, yellow, and red serves to signify additional statuses within the assembly process. These colors provide intuitive visual cues for tasks that are completed, awaiting action, or requiring attention, thereby enhancing the overall clarity and effectiveness of process monitoring.

Shapes can be connected and highlighted similarly with BPMN. This approach aligns with principles of re-use and iterative development, leveraging standardized interfaces and model editors. It allows for monitoring other assembly processes or general processes from the cobot with simpler modifications to the models.

5 CONCLUSION

In conclusion, digital twins offer immense potential to design, plan, optimize, and monitor systems in real-time throughout their lifecycle. The development of a model-based digital twin allows for the replication of physical system characteristics within a virtual environment, similar to traditional digital twins. However, our approach focuses on leveraging model-based techniques to extend the capabilities of digital twins for collaborative robots (cobots), providing the ability to reuse design artifacts. This promotes the iterative

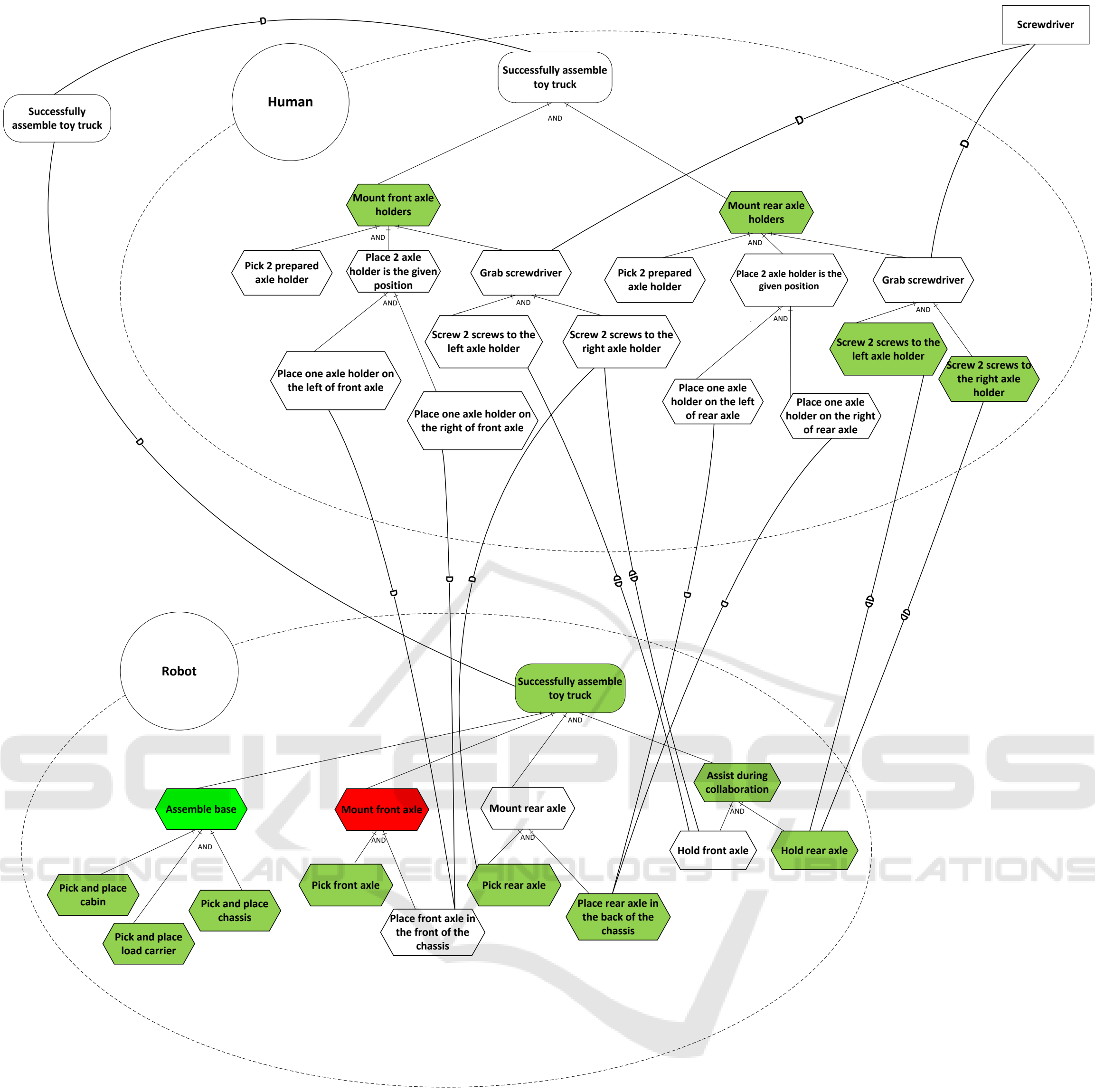

Figure 5: GRL Goal Model for the truck use case with highlighting.

and continuous development of the system over time.

A key advantage of model-based digital twins is their ability to enhance understanding during the early stages of development. By facilitating the early detection of potential issues, they enable proactive safety monitoring, improving both efficiency and reliability. In this paper, we introduced a model-based digital twin framework that utilizes GRL (Goal-oriented Requirement Language) goal models and BPMN (Business Process Model and Notation). Our approach emphasizes reusability by integrating standard interfaces of cobots and existing industrial systems, as well as employing widely accepted model editors. This ensures efficient integration, reducing development time and cost while leveraging current technologies and frameworks.

The contributions of this paper pave the way for the development of a model-based digital twin that can be directly implemented in industrial settings. As part of our future work, we will focus on creating conceptual models using standard model editors and evaluating the system with our monitoring implementation. In addition, we plan to apply different workflows and conduct in-depth safety analyses to further validate the system's capabilities.

Moving forward, we aim to enhance the functionality of our model-based digital twin by incorporating advanced features such as goal reasoning, which would allow the system to autonomously adjust based on evolving objectives. We also intend to explore modeling pattern analysis, enabling more efficient reuse of design artifacts across different applications. These enhancements will not only improve the twin's adaptability but also contribute to the broader adoption of digital twins in complex industrial environments.

REFERENCES

Bergs, T., Gierlings, S., Auerbach, T., Klink, A., Schraknepper, D., and Augspurger, T. (2021). The concept of digital twin and digital shadow in manufacturing. *Procedia CIRP*, 101:81–84.

Bruno, G. and Antonelli, D. (2018). Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells. *Int. Journal Advanced Manufacturing Technology*, 98:2415–2427.

Cañas, H., Mula, J., Díaz-Madroñero, M., and Campuzano-Bolarín, F. (2021). Implementing industry 4.0 principles. *Computers & industrial engineering*, 158:107379.

Das, A. N., Popa, D. O., Ballal, P. M., and Lewis, F. L. (2009). Data-logging and supervisory control in wireless sensor networks. *Int. Journal of Sensor Networks*, 6(1):13–27.

Daun, M., Brings, J., Krajinski, L., Stenkova, V., and Bandyszak, T. (2021). A grl-compliant istar extension for collaborative cyber-physical systems. *Requirements Engineering*, 26(3):325–370.

Daun, M., Manjunath, M., and Jesus Raja, J. (2023). Safety analysis of human robot collaborations with grl goal models. In *International Conference on Conceptual Modeling*, pages 317–333. Springer.

Daun, M., Stenkova, V., Krajinski, L., Brings, J., Bandyszak, T., and Weyer, T. (2019). Goal modeling for collaborative groups of cyber-physical systems with grl: reflections on applicability and limitations based on two studies conducted in industry. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1600–1609.

Davis, A. M. (1993). *Software requirements: objects, functions, and states*. Prentice-Hall, Inc.

Drath, R. (2021). *Automationml: A practical guide*. Walter de Gruyter GmbH & Co KG.

ITU Int. Telecommunication Union (2018). Recommendation itu-t z.151: User Requirements Notation (URN). Technical report.

Jesus Raja, J., Manjunath, M., and Daun, M. (2024). Towards a goal-oriented approach for engineering digital twins of robotic systems. In *ENASE*, pages 466–473.

Khan, F., Rathnayaka, S., and Ahmed, S. (2015). Methods and models in process safety and risk management: Past, present and future. *Process safety and environmental protection*, 98:116–147.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., and Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline*, 51(11):1016–1022.

Manjunath, M., Raja, J. J., and Daun, M. (2024). Early model-based safety analysis for collaborative robotic systems. *IEEE Transactions on Automation Science and Engineering*.

Mussbacher, G. and Nuttall, D. (2014). Goal modeling for sustainability: The case of time. In *2014 IEEE 4th Int. Model-Driven Requirements Engineering Workshop (MoDRE)*, pages 7–16. IEEE.

Petrasch, R. and Hentschke, R. (2016). Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method. In *2016 13th Int. Joint Conf. on Computer Science and Software Engineering (JCSSE)*, pages 1–5. IEEE.

Ramasubramanian, A. K., Mathew, R., Kelly, M., Hargaden, V., and Papakostas, N. (2022). Digital twin for human–robot collaboration in manufacturing: Review and outlook. *Applied Sciences*, 12(10):4811.

Redelinghuys, A., Basson, A. H., and Kruger, K. (2020). A six-layer architecture for the digital twin: a manufacturing case study implementation. *Journal of Intelligent Manufacturing*, 31(6):1383–1402.

Sandkuhl, K. and Stirna, J. (2020). Supporting early phases of digital twin development with enterprise modeling and capability management: Requirements from two industrial cases. In *Int. Conf. Business-Process and Information Systems Modeling, BPMDS*, pages 284–299. Springer.

Tantik, E. and Anderl, R. (2017). Integrated data model and structure for the asset administration shell in industrie 4.0. *Procedia Cirp*, 60:86–91.

Tao, F., Xiao, B., Qi, Q., Cheng, J., and Ji, P. (2022). Digital twin modeling. *Journal of Manufacturing Systems*, 64:372–389.

VanDerHorn, E. and Mahadevan, S. (2021). Digital twin: Generalization, characterization and implementation. *Decision support systems*, 145:113524.

Zhang, L., Zhou, L., and Horn, B. K. (2021). Building a right digital twin with model engineering. *Journal of Manufacturing Systems*, 59:151–164.