Towards a Goal-Oriented Approach for Engineering Digital Twins of Robotic Systems

Jeshwitha Jesus Raja[a], Meenakshi Manjunath[b] and Marian Daun[c]

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

Keywords: Requirements Engineering, Goal Modeling, Digital Twin, Industry Automation, Robotic Systems.

Abstract: In many smart manufacturing scenarios of Industry 4.0, robots play a vital role. Robotic systems allow for automatization and semi-automatization of individual work tasks using standard hardware. Thus, production and assembly processes can be flexibly redefined during operation. In addition, human workers can be supported for complex and specific work tasks where full automation by industrial production systems is not possible or not cost-efficient. To monitor current process execution, to predict process outcome, and to ensure safe behavior of the robots at runtime, digital twins are seen as a vital part of future smart manufacturing. However, current industrial approaches typically define the digital twin on the go, i.e. when the factory has been build and equipped with robotic systems. Thus, the absence of systematic planning of the digital twin leads to unused potential for more complex analysis, monitoring, and prediction tasks of digital twins commonly suggested in research. This is partly due to the absence of structured software and systems engineering approaches for the development of robotic systems. In this paper, we explore the use of goal modeling to systematically define the robotic system, its monitoring system, and the digital twin. Application to case examples shows that this lightweight approach aligns with industry preferences to focus on technical challenges rather than invest too much effort in a thorough yet cost intensive engineering approach, while at the same time allowing for the proper definition of robots and their digital twins.

1 INTRODUCTION

Robotic systems involve mechanisms that interact with their surroundings, including humans, utilizing an array of sensors, actuators, and interfaces to offer intelligent services and information (Demir and Turan, 2021). Robot systems must work to achieve tasks while monitoring and reacting to unexpected situations (Kortenkamp et al., 2016). Digital twins are virtual representations of physical systems (Koulamas and Kalogeras, 2018). Within smart manufacturing, incorporating robotic systems for production, digital twins are used to monitor and analyze the status of their physical counterparts. Consequently, digital twins aid in planning real-time adaptations by detecting unexpected events or errors during process execution.

Among others, the huge potential of digital twins is seen in supporting safety assurance at runtime, exploring problem spaces to find optimal or near optimal solutions (Liu et al., 2020), foster prediction of runtime properties (Tao et al., 2019), foster more sustainable manufacturing (Xu et al., 2022), or improve cybersecurity (Bécue et al., 2022). Thus, the basic idea is that the twin can be used to simulate the execution of a task in a safe environment, and, as a consequence, the actual robot will only execute this task if determined safe by the digital twin (Kor et al., 2023). This also allows a better instrumentation for monitoring system execution and determining the current state of the system (Tao et al., 2019).

[a] https://orcid.org/0009-0008-7886-7081
[b] https://orcid.org/0009-0005-6421-1450
[c] https://orcid.org/0000-0002-9156-9731

1.1 Current Challenges in the Definition of Digital Twins for Robotic Systems

The utilization of digital twins in the field of robotic systems is relatively new and presents several challenges. Some of the most common ones include compatibility issues, high development costs, uncertainty in development and real-time feedback, as well as low accuracy and precision. This necessitates the acquisition of new sensory devices that are more compatible, consequently leading to an increase in the overall

Jesus Raja, J., Manjunath, M. and Daun, M.
Towards a Goal-Oriented Approach for Engineering Digital Twins of Robotic Systems.
DOI: 10.5220/0012681500003687

In *Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2024)*, pages 466-473
ISBN: 978-989-758-696-5; ISSN: 2184-4895

cost (Ramasubramanian et al., 2022). Notably, the flexibility of human-robot collaborations expands the range of conceivable states in the digital twin. Furthermore, in cyber-physical systems, challenges related to connectivity, computational power, and various other factors significantly impact the creation of a highly reliable digital twin (Ding et al., 2019).

Albeit the existence of these technical challenges, there exists a need for structured development approaches, systematically designing the digital twins in combination with their physical counterparts. There particularly exist a need for approaches supporting early development phases (Sandkuhl and Stirna, 2020). A promising approach to overcome this lack of systematic engineering approaches is seen in model-based development, which is popular in the context of industry 4.0 (Wortmann et al., 2020).

1.2 Contribution

Goal modeling is an established lightweight modeling approach for early development phases (Van Lamsweerde, 2001; Horkoff et al., 2019). In this paper, we explore the use of goal modeling with the goal oriented requirement language (GRL, (ITU Int. Telecommunication Union, 2018)) for systematically defining the robotic system, its monitoring system, and the digital twin. Application to an industry example shows that this lightweight approach aligns with industry preferences to focus on technical challenges rather than invest too much effort in a thorough yet cost intensive engineering approach, while at the same time allowing for the proper definition of robots and their digital twins. Additionally, the utilization of runtime analysis through goal reasoning unveils more benefits in understanding the system and its digital twin.

1.3 Outline

The paper is outlined as follows. Section 2 gives an overview of the related work. Section 3 presents the goal-oriented engineering approach for developing robotic systems and their digital twins, being used to explore the use of goal modeling in the engineering of these systems. Based on this approach, Section 4 evaluates the applicability and usefulness of this approach using an industry case example from the industry automation domain. Finally, Section 5 concludes the paper and discusses the next steps in future work.

2 RELATED WORK

Goal modeling is an established approach in requirements engineering (Van Lamsweerde, 2001; Horkoff et al., 2019). The concept of goal models has been advocated to express stakeholder objectives and to capture and choose among requirement alternatives (Horkoff and Yu, 2016). Goal models have proven useful for eliciting, documenting, and validating stakeholder intentions (Van Lamsweerde, 2001). They are commonly used in early phase requirements engineering to document the basic high-level requirements and to already reason over fundamental design decisions and to identify crucial conflicts in the very early stages of development (Grubb and Chechik, 2021).

The most common goal modeling approaches are KAOS (Dardenne et al., 1993; Van Lamsweerde, 2009) and iStar (Yu, 1997; Dalpiaz et al., 2016). The iStar (originally i*) framework defines a conceptual modeling language for capturing and analyzing properties of complex systems in terms of actors, their intentions, and their relationships (Amyot et al., 2009). The Goal-oriented Requirement Language (GRL) is a lightweight standardized version of iStar, it is regulated by the International Telecommunication Union (ITU) in its recommendation Z.151 (ITU Int. Telecommunication Union, 2018). The GRL documents goals in graph-based structures. Actors are used to define the belonging of goals to different stakeholders or, as in the case of this paper, to systems. In addition to goals, the GRL defines further intentional elements that allow separating different concepts like qualities and tasks. Intentional elements can be decomposed, and they can contribute to each other, or even depend on each other.

Analyses (i.e. goal satisfaction analysis or reasoning) of goal models allow for early detection of defects (Giorgini et al., 2003; Brings et al., 2019). Thereby, the benefit of goal models is often seen in early discovery and definition of relations (e.g., contributions, dependencies, conflicts) between different requirements (Kavakli, 2004), as goal reasoning can also be used to allow complex analyzes in the early development phases (Pardillo and Trujillo, 2008).

3 APPROACH

3.1 Overview and Process Steps

The major idea of the approach is to use a goal model for runtime analysis, as has also been suggested by (Cheng et al., 2014). It suggests that goals can provide assurance at runtime as they adapt to changes in their execution environment. Goal modeling enables assurance techniques, and modification of the model to adapt to the change of the application.

For the digital twin, we want to foster re-use of design time models. As has been shown by Daun et al. (Daun et al., 2019), GRL goal models are a good approach to define requirements and analyze early design decisions for cyber-physical systems, particular in the domain of industry automation. Therefore, we aim at re-using design time models for the smart factory, especially for the runtime analysis. In particular, we build upon a iStar-compliant GRL extension (Daun et al., 2021), which we have shown to be applicable to robotic production systems (Daun et al., 2023). In previous work, we have shown that goal models can be used to specify the digital twin (Jesus Raja et al., 2023), in this paper we focus on the systematic development of the digital twin goal model based on the goal model of the robotic system. Figure 1 shows the overall idea.

First, the smart factory and the individual cyber-physical production and transport systems of the smart factory are defined using a GRL goal model. Based on the goal model, the digital twin is defined. The production system is completely automated and therefore only requires humans as a backup monitoring system. Therefore, the initial goal model is extended with monitoring tasks reflecting the runtime objectives of the digital twin itself. Finally, the digital twin goal model is linked to the factory and, based on monitoring data, the system is updated to reflect the current status of the factory. We then use, goal reasoning techniques to identify problematic situations and propose runtime adaptations.

In summary – and as outlined in Figure 1, we propose the goal-oriented specification of the physical and the digital twin with four consecutive process steps.

- **Step 1: Specify the Physical Twin (i.e. the System) in a Goal Model.** First, a goal model is created defining the physical twin. This goal model focuses on the goals to be achieved by each robot to be part of the overall robotic system. For these goals, tasks are defined to specify the basic functionality needed for each robot.
- **Step 2: Specify the Monitoring System by Extending the System Goal Model.** Robotic systems heavily rely on a wide range of monitoring devices. These can be, for instance, cameras needed to identify certain poses of work pieces or the robot itself, or even safety shutdown mats monitoring whether a human worker is in a certain area coming too close to the robot. In addition, monitoring devices need to be specified for each task from Step 1. This allows representing the current state of the robot (and its goal fulfillment) in the digital twin.
- **Step 3: Specify the digital twin and Combine the Goal Models of the Different Systems and their Monitoring Systems.** In this step, the goal model of the digital twin is defined. In addition, to the simple monitoring tasks, the digital twin shall typically provide some additional business value. For instance, the digital twin shall predict whether a potential safety hazard might occur in the direct future (i.e. if a worker is likely to step into the path of the robot), whether the current assembly process might fail due to insufficient adherence to the specified process execution, or the tool needs replacement due to wearing. In addition, often a robotic system – specifically in manufacturing and assembly processes – consists of multiple individual robots. In this case, the digital twin needs to aggregate the data collected from the different monitoring systems and the different goal models of each robot need to be combined.
- **Step 4: Specify Goal Fulfillment Criteria for Runtime Analysis.** Finally, the goal model shall be used as a runtime model to highlight the current state of goal fulfillment for the robot and the monitoring system (e.g., to detect system failures) and to foster the prediction of the outcome of the current production step. Therefore, goal fulfillment criteria need to be defined for each task of the system and the monitoring system.

Furthermore, Figure 1 shows some needed iterations. In our case, we follow a bottom-up approach as it best fits the thought process of current industrial engineers (i.e. define first the robot you want to build and later think about the digital twin). However, in this process, it is not ensured that all tasks needed for functionality of the digital twin has already been defined. Therefore, the definition of the digital twin model will often lead to the need to revise the monitoring system model, or even the system model. When defining the monitoring system in the first place, emphasis is often given to a) the definition of monitoring devices needed for proper system execution (e.g., pose recognition), b) the definition of monitoring devices for safety concerns (e.g., detect human workers in a certain area), or c) the implementation of quality assurance measures (e.g., check that the work product has been assembled correctly). Thus, when defining the desired functionality of the digital twin, it will often be detected that more monitoring devices are needed to support predictions. This can also be monitoring devices not directly attached

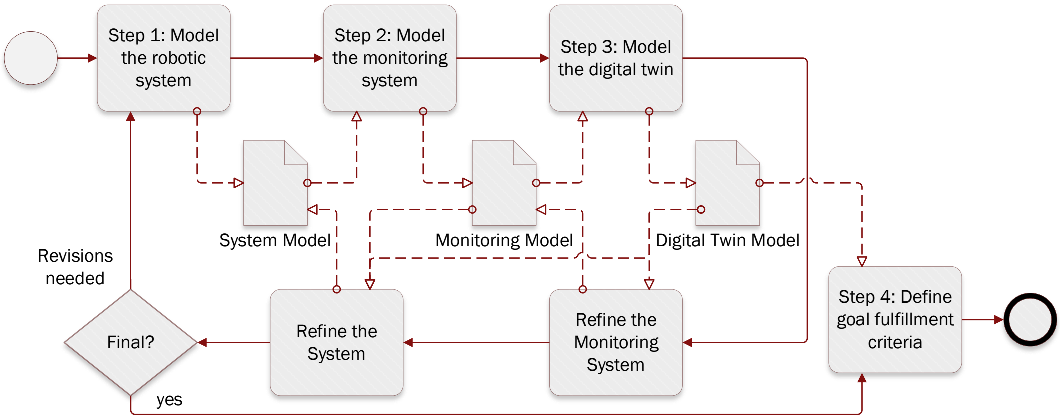

Figure 1: Overview.

to the physical twin. For instance, it might be necessary to monitor the production rate of a certain work piece produced by another machine in the factory to estimate whether the current assembly process might be disturbed.

3.2 Modeling the System

We use our goal modeling extension for collaborative cyber-physical systems to model the physical twin. In addition, we restrict the way the goal model is created to foster the later development of the digital twin. Therefore, we need to ensure that the goal model can be properly connected to the real world. This means that the elements of the goal model must be monitored. This allows updating goal fulfillment of the model according to the current state of the system and use reasoning techniques to analyze the system.

Important for linking the goal model to the real world are tasks. As our primary focus lies in monitoring our data, 'tasks' serve as a means of monitoring as they articulate the system's behavior. Tasks represent abstract instructions executed by a system to attain a goal (ITU Int. Telecommunication Union, 2018). For a more comprehensive understanding of functionality, tasks can be further broken down into finer-grained tasks. In production processes, task fulfillment is determined by monitoring the outcomes of the tasks. This modeling approach is in-line with existing ideas to foster a more structured approach to modeling GRL and iStar goal models (Keller et al., 2018).

Figure 2 shows this goal modeling approach for a collaborative robot (cobot) used in a manufacturing production line for welding a car's doors to its chassis. The cobot shown is responsible for executing a simple pick and place operation. This is an operation commonly needed in robotic automation as the work pieces needed, must be identified, selected, and positioned properly, so that subsequent robots (and partly human workers) can use them.

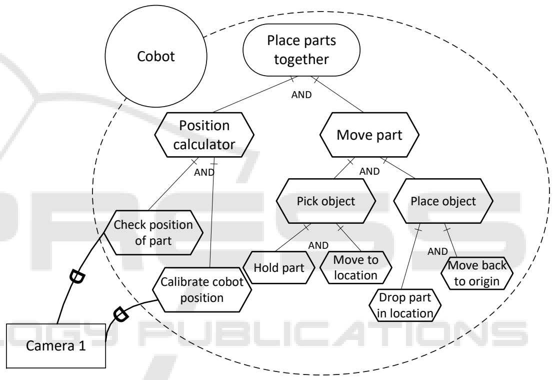

Figure 2: Goal model of the physical twin.

The goal model illustrates all the tasks the cobot must execute to pick and place the parts necessary for welding a car. The cobot initiates the process by calculating and calibrating the positions of the parts required for picking. These procedures depend on a camera, which is considered an external resource in our goal model as it does not belong to the cobot but to the factory itself. Subsequently, the cobot proceeds to move the picked parts to their designated positions. The essential sub-tasks, 'pick object' and 'place object,' play a pivotal role in accomplishing the overarching task of 'Move part.' Both of these sub-tasks have their own respective sub-tasks that need completion for the entire process to succeed. All tasks and sub-tasks are interconnected through an 'AND' connection, signifying that the successful completion of the process hinges on each individual task and subtask being completed without any errors.

3.3 Modeling the Monitoring System

Subsequently, we define the goal model for the digital twin by extending the existing goal model of the cobot (Figure 2) to incorporate essential elements of the monitoring system that are later used by the digital twin. The resulting goal model for the monitoring system is depicted in Figure 3. Notably, we have introduced monitoring tasks dedicated to overseeing system execution. These are represented as distinct goals connected based on dependencies.

To ensure system monitoring during operation, multiple sensors and cameras are strategically positioned. Various potential system malfunctions — such as sensor errors, human errors, part misalignment, interference from other equipment, or environmental factors — are anticipated. These refined monitoring tasks are intricately tied to the cobot's process tasks, enabling clear associations between monitoring and the specific processes like pick and place in our case. Additionally, we're able to introduce supplementary monitoring tasks unrelated to the primary process tasks. For instance, one task involves evaluating the work product after completing its task of calculating the product's position, while another focuses on precisely picking and placing the product. This latter task aims to verify whether the calibration aligns with the digital twin's predictions. Notably, a second camera is utilized for this monitoring task, distinct from the task that checks the part's position. Here, the objective isn't solely to monitor task accuracy; rather, it's to confirm whether the calibration aligns with the predicted results.

3.4 Modeling the Digital Twin

In the next step, the digital twin can be defined for each robotic system or for the system-of-systems (i.e. for all robotic systems together). Therefore, we again use GRL goal modeling and integrate a new actor for the digital twin into the goal model. Thereby, we can differentiate between the physical system, the monitoring of the system, and the analysis procedures of the digital twin. The goals and tasks of the digital twin are then connected to the monitoring tasks. Thereby, it is ensured that all analysis procedures are sufficiently grounded in the available data gathered by the monitoring system during runtime. In this phase, it is likely that data is identified that is needed for the digital twin but has not yet been monitored. Therefore, the goal model of the monitoring system will need revision and integrate these new monitoring tasks. Another likely effect is the introduction of the functionality in the system due to the outcome of predictions. As the results of the calculation conducted by the digital twin shall impact the physical twin (e.g., to optimize process execution, to stop unsafe execution), functionality is needed in the system that allows reacting to the outputs of the digital twin.

3.5 Goal Reasoning for Runtime Analysis

Using goal modeling to model the system and the digital twin, might allow us to reason over the goal model to identify goal fulfillment or the lack of goal fulfillment during runtime. Therefore, we investigate different application scenarios for existing goal satisfaction analysis approaches to support runtime analysis and adaptation planning of the digital twin.

For the monitoring, forward reasoning (Amyot et al., 2010) can be applied, thereby, we need to ensure that all leaves of the goal graph are monitored. As discussed above, the goals of the cobot have been decomposed until reaching a level of fine-grained tasks that can be monitored. Subsequently, the monitoring data is used to estimate whether a task is fulfilled or not. Based on these measurements, the goal fulfillment is propagated up to the higher level goals. Thus, can be determined if the cobot is currently performing its duties as intended or whether an intervention is needed.

A basic distinction commonly made for reasoning approaches is the differentiation between qualitative reasoning and quantitative reasoning (Amyot et al., 2010; Giorgini et al., 2003).

- Qualitative Reasoning. The starting goals (i.e. the leaf goals in forward reasoning and the upper goals in backward reasoning) are assigned goal satisfaction labels like, e.g., 'satisfied', 'weakly satisfied', 'weakly denied', 'denied'. Then different propagation rules are applied to determine how these are propagated. For instance, an AND-decomposed 'satisfied' and 'weakly satisfied' will be composed to 'weakly satisfied'.
- Quantitative Reasoning. Instead of goal satisfaction labels, numeric values are used and propagated. While different scales are possible, a common approach is using values from '0' to '100', being interpreted as percentage values. Based on mathematical formulas used as propagation rules, these values are then added, averaged, subtracted, etc. to propagate goal fulfillment.

We found both approaches – qualitative and quantitative reasoning – applicable for runtime analysis. Both approaches can be used to support runtime monitoring of system execution. The propagation of goal

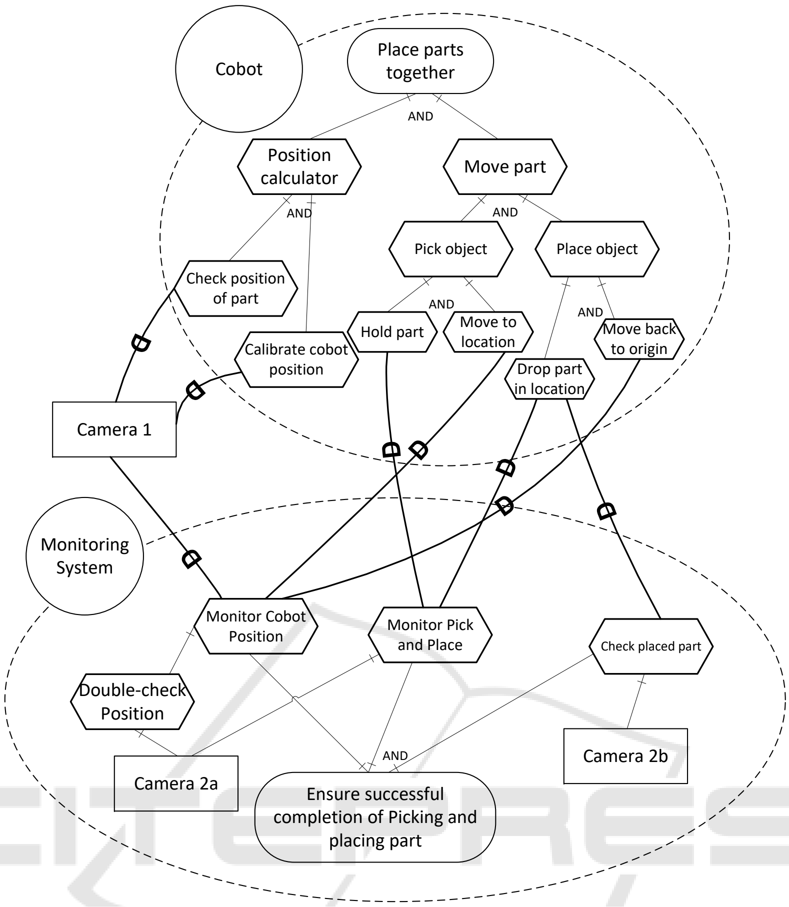

Figure 3: Goal model of the monitoring system.

satisfactions helps in identifying which parts of the system function correctly or need support at runtime. Defining thresholds for the quantitative approach poses a challenge. We need to be able to determine when we consider the system working correctly. Due to sensor imprecision, blocked views of cameras, etc. it is highly unlikely to reach 100% goal satisfaction. Therefore, we need to define a threshold like, with a goal satisfaction probability of 95% we assume this goal to be fulfilled.

Determining goal satisfaction of the leaf tasks pose an even more challenging situation. The question of when do we consider a task to be satisfied or not, is difficult to answer. Problematic is the use of 'weakly satisfied' or 'weakly denied', which we therefore discourage. The same obviously holds true for quantitative reasoning. It is a challenge to determine, when can we reach 97% goal satisfaction for the welding task or 76% goal satisfaction for a monitoring task. Despite this difficulty in application, this needs to be done either way. Even traditional monitoring approaches work with probabilities and thresholds. Thus, we see the great benefit of a goal-oriented approach in having this discussion early during system design. Right at the moment of the development of the digital twin, we need to determine how we can instantiate concrete determination of goal satisfaction for the various values we measure.

4 EVALUATION

We evaluated the use of GRL goal models to develop robotic systems and their digital twins using a case study from the industry automation domain. We chose an industrial production and assembly line for this. Overall, we can state that the use of GRL goal models is a fitting approach to model robotic systems and support the definition of the digital twin. Particularly, they offer support for identifying the tasks to be monitored and defining the means to do so. For instance, in our case, each monitoring system uses an independent camera to allow detecting camera miscalibrations of the primary camera used by the respective cobot. In a next step, analysis of the model and identifying overlaps between the different monitoring systems can result in finding further quality checking approaches by comparing the overlapping monitoring output of the individual monitoring systems.

We found the following major benefits for using GRL goal models in the development of digital twins in industry automation:

- Re-use. Requirements goal models can be re-used twice. First for defining the digital twin. Second, for runtime monitoring and analysis. Therefore, the goal models can easily be extended by monitoring tasks responsible for monitoring the production tasks of the individual cobots. Thus, it also allows for a more continuous development of a system, as changes in the factory outlet are incorporated into the goal model and thus directly reflected in the requirements.
- Analysis. Forward and backward reasoning approaches for goal models can be easily applied to support monitoring of the factory state (i.e., by up-propagating runtime goal fulfillment) and calculating interventions (i.e., by down-propagating goal fulfillment to determine necessary actions).
- Simplicity for human-in-the-loop use cases. Smart factories are commonly monitored and run by a factory manager. This human has the task to ensure safe functioning of the factory and is involved in re-scheduling and optimization tasks. Therefore, a simple graphical model highlighting dependencies between different tasks is beneficial.

Finally, we need to briefly line out the major limitations. As common for case study research, we have only gathered insights for one case example, thus, generalizability cannot be assumed. Furthermore, comparative experimental research is needed to determine whether using goal satisfaction analysis at runtime can actually contribute to digital twins in industrial production systems. Additionally, investigating the potential of using goal satisfaction analysis at runtime needs more thorough consideration in the future, particularly considering the definition of thresholds and providing support for systematically defining goal satisfaction labels or percentages based on monitoring data.

5 CONCLUSION

In this paper, we investigated the use of GRL goal models to develop robotic systems and their digital twins. GRL goal models support easy specification of high-level requirements and their interrelations. The digital twin allows for in-depth real time analysis of the current state of the factory and supports action planning for new tasks or evaluating error handling strategies for defects occurring at runtime. For the future of manufacturing, digital twins play a vital role.

We have shown that GRL goal models are applicable to model not only the cyber-physical production systems within a factory, but also the needed monitoring systems belonging to the digital twin. This way, the digital twin can be developed closely linked to the individual goals of the systems to be developed. In consequence, it can be ensured that every major production step – which is defined as task contributing to a major goal of a production system – is monitorable and will be monitored by the digital twin. The use of goal models furthermore supports the definition of thresholds and decision-making about how to define that a process step is actually executed correctly, and the corresponding task can be considered fulfilled. Based on these definitions, goal satisfaction analyses can be applied to estimate the overall goal fulfillment of the factory and to calculate adaptation scenarios to reach goal fulfillment. In addition, we identified some modeling patterns for digital twins for industrial manufacturing.

Future work, will have to validate these modeling patterns in other scenarios. Furthermore, a structured approach to estimate goal fulfillment based on monitoring data will help goal-based development of digital twins.

REFERENCES

Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., and Yu, E. (2010). Evaluating goal models within the goal-oriented requirement language. *Int. Journal of Intelligent Systems*, 25(8):841–877.

Amyot, D., Horkoff, J., Gross, D., and Mussbacher, G. (2009). A lightweight grl profile for i* modeling. In *Advances in Conceptual Modeling-Challenging Perspectives: ER 2009 Workshops*, pages 254–264. Springer.

Bécue, A., Praddaude, M., Maia, E., Hogrel, N., Praça, I., and Yaich, R. (2022). Digital twins for enhanced resilience: Aerospace manufacturing scenario. In *Int. Conf. on Advanced Information Systems Engineering*, pages 107–118. Springer.

Brings, J., Daun, M., Bandyszak, T., Stricker, V., Weyer, T., Mirzaei, E., Neumann, M., and Zernickel, J. S. (2019). Model-based documentation of dynamicity constraints for collaborative cyber-physical system architectures: Findings from an industrial case study. *Journal of systems architecture*, 97:153–167.

Cheng, B. H., Eder, K. I., Gogolla, M., Grunske, L., Litoiu, M., Müller, H. A., Pelliccione, P., Perini, A., Qureshi, N. A., Rumpe, B., et al. (2014). Using models at runtime to address assurance for self-adaptive systems. *Models@ run. time: foundations, applications, and roadmaps*, pages 101–136.

Dalpiaz, F., Franch, X., and Horkoff, J. (2016). istar 2.0 language guide. *arXiv preprint arXiv:1605.07767*.

Dardenne, A., Van Lamsweerde, A., and Fickas, S. (1993). Goal-directed requirements acquisition. *Science of computer programming*, 20(1-2):3–50.

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 *ER 2023: Int. Conf. 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 *34th ACM/SIGAPP Symp. on Applied Computing*, pages 1600–1609.

Demir, K. A. and Turan, B. (2021). Developing trends in power and networking technologies for intelligent cities. In *Developing and monitoring smart environments for intelligent cities*, pages 61–85. IGI Global.

Ding, K., Chan, F. T., Zhang, X., Zhou, G., and Zhang, F. (2019). Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors. *Int. Journal of Production Research*, 57(20):6315–6334.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E., and Sebastiani, R. (2003). Reasoning with goal models. In *ER 2002: 21st Int. Conf. on Conceptual Modeling*, pages 167–181. Springer.

Grubb, A. M. and Chechik, M. (2021). Formal reasoning for analyzing goal models that evolve over time. *Requirements Engineering*, 26(3):423–457.

Horkoff, J., Aydemir, F. B., Cardoso, E., Li, T., Maté, A., Paja, E., Salnitri, M., Piras, L., Mylopoulos, J., and Giorgini, P. (2019). Goal-oriented requirements engineering: an extended systematic mapping study. *Requirements engineering*, 24:133–160.

Horkoff, J. and Yu, E. (2016). Interactive goal model analysis for early requirements engineering. *Requirements Engineering*, 21:29–61.

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

Jesus Raja, J., Manjunath, M., Kranz, P., Schirmer, F., and Daun, M. (2023). Using goal modeling for defining digital twins in industry automation. In *Companion Proceedings 42nd Int. Conf. Conceptual Modeling*.

Kavakli, E. (2004). Modeling organizational goals: Analysis of current methods. In *ACM Symp. on Applied computing*, pages 1339–1343.

Keller, K., Brings, J., Daun, M., and Weyer, T. (2018). A comparative analysis of itu-msc-based requirements specification approaches used in the automotive industry. In *10th Int. Conf. System Analysis and Modeling*, pages 183–201. Springer.

Kor, M., Yitmen, I., and Alizadehsalehi, S. (2023). An investigation for integration of deep learning and digital twins towards construction 4.0. *Smart and Sustainable Built Environment*, 12(3):461–487.

Kortenkamp, D., Simmons, R., and Brugali, D. (2016). Robotic systems architectures and programming. *Springer handbook of robotics*, pages 283–306.

Koulamas, C. and Kalogeras, A. (2018). Cyber-physical systems and digital twins in the industrial internet of things [cyber-physical systems]. *Computer*, 51(11):95–98.

Liu, C., Jiang, P., and Jiang, W. (2020). Web-based digital twin modeling and remote control of cyber-physical production systems. *Robotics and computer-integrated manufacturing*, 64:101956.

Pardillo, J. and Trujillo, J. (2008). Integrated model-driven development of goal-oriented data warehouses and data marts. In *ER 2008: 27th Int. Conf. on Conceptual Modeling*, pages 426–439. Springer.

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.

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 *Enterprise, Business-Process and Information Systems Modeling: 21st Int. Conf. BPMDS, 25th Int. Conf. EMMSAD*, pages 284–299. Springer.

Tao, F., Qi, Q., Wang, L., and Nee, A. (2019). Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. *Engineering*, 5(4):653–661.

Van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. In *Fifth ieee Int. Symp. on requirements engineering*, pages 249–262. IEEE.

Van Lamsweerde, A. (2009). *Requirements engineering: From system goals to UML models to software*, volume 10. Chichester, UK: John Wiley & Sons.

Wortmann, A., Barais, O., Combemale, B., and Wimmer, M. (2020). Modeling languages in industry 4.0: an extended systematic mapping study. *Software and Systems Modeling*, 19:67–94.

Xu, L., de Vrieze, P., Lu, X., and Wang, W. (2022). Digital twins approach for sustainable industry. In *Int. Conf. on Advanced Information Systems Engineering*, pages 126–134. Springer.

Yu, E. S. (1997). Towards modelling and reasoning support for early-phase requirements engineering. In *ISRE'97: 3rd IEEE Int. Symp. on Requirements Engineering*, pages 226–235. IEEE.