



# Self-improving situation awareness for human–robot-collaboration using intelligent Digital Twin

Manuel Müller<sup>1</sup> · Tamás Ruppert<sup>2</sup> · Nasser Jazdi<sup>1</sup> · Michael Weyrich<sup>1</sup>

Received: 23 November 2022 / Accepted: 19 April 2023 / Published online: 24 May 2023  
© The Author(s) 2023

## Abstract

The situation awareness, especially for collaborative robots, plays a crucial role when humans and machines work together in a human-centered, dynamic environment. Only when the humans understands how well the robot is aware of its environment can they build trust and delegate tasks that the robot can complete successfully. However, the state of situation awareness has not yet been described for collaborative robots. Furthermore, the improvement of situation awareness is now only described for humans but not for robots. In this paper, the authors propose a metric to measure the state of situation awareness. Furthermore, the models are adapted to the collaborative robot domain to systematically improve the situation awareness. The proposed metric and the improvement process of the situation awareness are evaluated using the mobile robot platform *Robotino*. The authors conduct extensive experiments and present the results in this paper to evaluate the effectiveness of the proposed approach. The results are compared with the existing research on the situation awareness, highlighting the advantages of our approach. Therefore, the approach is expected to significantly improve the performance of cobots in human–robot collaboration and enhance the communication and understanding between humans and machines.

**Keywords** Situation awareness · Intelligent Digital Twin · Collaboration · Metrics

## Introduction

Close cooperation between humans and collaborative robots (cobots) is envisioned to make future production particularly efficient by combining the strengths of humans and machines and compensating for their weaknesses. To achieve this, cobots must master complex problems in changing environments. Accordingly, the models need to be constantly

updated. In this way, the cobot must learn to reassess situations and constantly adapt behavior. A summary of the key considerations in the acquisition and deployment of cobots is proposed in Cohen et al. (2022). However, this change has the potential for misunderstanding with the worker with whom the robot interacts. In fact, the main issue of the cobot is the human factors, as the cobot considers three main dimensions: robot features, modern production systems characteristics, and human factors (Faccio et al., 2022). On the other hand, the lack of predictability of the cobot's actions, and the doubt whether the cobot understands the current situation correctly and acts responsibly can be a source of psychological stress for the operator. When discussing human-machine collaboration, the human feeling towards the automation system must be considered (Azni Jafar et al., 2014). The concept of Operator 4.0 (Romero et al., 2016) focuses on supporting the human operators with the enabling technologies (Ruppert et al., 2018). The Cognitive Operator 4.0 proposes a deep perception, awareness, and understanding between both collaborative agents (Thorvald et al., 2021). To this end, a connecting link is required: situation awareness. While the situation awareness of the human operator has been studied

---

✉ Manuel Müller  
manuel.mueller@ias.uni-stuttgart.de

Tamás Ruppert  
ruppert@abonyilab.com

Nasser Jazdi  
nasser.jazdi@ias.uni-stuttgart.de

Michael Weyrich  
michael.weyrich@ias.uni-stuttgart.de

<sup>1</sup> Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Baden-Württemberg, Germany

<sup>2</sup> ELKH-PE Complex Systems Monitoring Research Group, Department of Process Engineering, University of Pannonia, Egyetem str. 10. Veszprém, Veszprém 8200, Hungary

(Endsley, 1995a), there is no equivalent concept for cobots. Therefore, cobot situation awareness may be the next essential element of human–robot collaboration.

To build the situation awareness for cobots, the first challenge is to measure it. In a dynamic environment, situations change quickly and the system needs to adapt. Just as humans have to act more carefully in new environments because they do not fully understand what is going on, cobots should do the same. But how will the cobot know that an environment is actually new? How will the cobot know how much its situation awareness has decreased and therefore how cautious it should act? There are approaches to improve models by parameter optimization (El Ouanjli et al., 2022), support vector regression (Xie et al., 2018), and so on. There are approaches using transfer learning to bridge the simulation-to-reality gap (Collins et al., 2020). There are also anomaly detection approaches that indicate when a system does not behave as the model predicts it Lindemann et al. (2021). However, none of these approaches provide a metric that measures how well the system understands the situation. The ability of the robot to reflect on its own understanding is key to the human operator building trust in the cobot and moving towards the Cognitive Operator 4.0. This leads to the research question:

*RQ1: How can situation awareness be measured for cobots?*

The next problem is about communicating this information. The upcoming technologies heavily rely on models and the capability of making sense of them. On top of this, the cobot needs to communicate these insights to the worker to create mutual understanding. This is particularly challenging because computer systems perceive the environment differently than humans and sometimes draw different conclusions. This is because both sensory and cognitive capabilities differ between humans and robots. A virtual representation is required to let the human workers dive into the insights the robot generates. This information must be provided in a human-understandable way to move from a system-centric view to a human-centric view. The latter challenge is formulated as part of the Operator 4.0 (Löcklin et al., 2021) and leads to the second research question:

*RQ2: How can the state of the situation awareness be communicated to a human worker using the intelligent Digital Twin?*

Having measured the situation awareness, the next step is to optimize for it. In this way, the cobot does not only adapt its behavior but also provides resilience. The European Union (EU) announced the 5th Industrial Revolution (Breque et al., 2021) to respond to the current issues of the manufacturing and the supply chain as Industry 5.0 defines three main pillars: (i) Sustainability, (ii) Resilience, and (iii) Human-centricity (European Commission, 2021). Operator 5.0 (Romero & Stahre, 2021) aims to solve the last two issues.

The existing approaches are capable of optimizing dedicated models or sets of models which they are engineered for. However, to the best of the authors' knowledge, a process that autonomously detects when an adaptation is required and efficiently corrects the model pattern in parallel with the operation and in a context-dependent manner does not yet exist. Although studied for the human worker (Endsley, 1995a), the process of increasing situational awareness has not yet been applied to cobots, leading to the following question:

*RQ3: How can a cobot undergo the situation awareness process of increasing situation awareness?*

To answer these research questions, the Digital Twin (DT) concept is a valuable foundation (Pairet et al., 2019). As the need for human-machine collaboration is particularly important in the aerospace domain, NASA launched the DT concept in 2012, defined as the “virtual representation of a physical asset” (Ashtari Talkhestani et al., 2018). In the context of this work, this virtual representation includes the modeling of the system's environment. Since its initiation in 2012, the concept of DT has evolved. The quality of the virtual representation depends directly on the quality of the models. However, the question of where to start and end modelling is still debated. According to West and Blackburn (2018), this quality of models competes with effort. On the one hand, it is impractical or at least uneconomical to model every detail. On the other hand, outdated or inaccurate models can lead to misinterpretation of a situation and thus to suboptimal or even dangerous patterns of action. This is where the DT needs intelligence to manage and communicate its models autonomously. To this end, the intelligent DT (iDT) Ashtari Talkhestani et al. (2018) extends the concept of the DT to include aspects of intelligence such as data analysis and reasoning. Situation consciousness, specifically environmental and self-consciousness, comes into play. Situation-consciousness represents the level of understanding and therefore the quality of awareness.

Consequently, a high situation consciousness correlates with the recognition of model boundaries, the synchrony of virtual and physical worlds (for humans, this is the gap between imagination and reality), and the identification and characterization of perturbation events. This work contributes to building situation consciousness with the following main contributions:

- The proposed metric allows for the measurement of situation awareness in the case of cobots. In this way, the cobot can reflect on its behavior depending on how familiar it is with the current situation. To this end, the need for changes in models can be uncovered at run-time.
- Based on the defined indicators, situation awareness can be improved using the developed iDT. This improvement process provides resilience and is sample efficient com-

pared to traditional reinforcement learning approaches, requiring only 10% of samples for training.

- The developed framework using the iDT demonstrates efficient communication between humans and robots. In this way, the approach contributes to more trust and efficiency in human–cobot collaboration. In particular, it supports dynamic risk and reliability assessment, as the reason for the changes are quantified in the consciousness metric and the changes are limited to edge cases, where the active models reach their limit.

The remainder of this article is organized as follows: The paper continues with the related works (“[Related works](#)” section). After that, the authors introduce the situation consciousness to answer the first research question (“[Situation-consciousness: the measurement of situation awareness for cobot](#)” section). “[The process of improving situational awareness](#)” section details the process of situation consciousness building. The subsequent “[Experiments and results](#)” section exemplarily shows the application of the situation consciousness and its improvement with experiments. The paper closes with some conclusions and future works (“[Conclusions and future Work](#)” section).

## Related works

*A Humanities Perspective on Consciousness and Awareness.* Consciousness is heavily discussed in the humanities such as neuroscience and psychology (Solms, 1997) as it describes the degree of understanding of what is happening to and around an individual. It is a way of assessing the state of awareness. For humans, there are different approaches to measuring consciousness (Irvine, 2013). Roughly speaking, the methods for determining the consciousness of the human group are either to ask the respondent to describe what he or she experienced (subjective method) or to measure neural activity in the brain (objective method). Unfortunately, these measurement methods need to be modified. To the best of the authors’ knowledge, there is no measurement of situation consciousness for the technical domain. However, researchers have studied situation awareness in the technical domain since the 1990s.

*Situation Awareness and its Measurement in the Technical Domain.* In the well-cited work by White (1991), the author locates the situation assessment in level 2 of the Joint Directors of Labs (JDL) sensor fusion model. In this level, “knowledge about objects, their characteristics, their relationships to each other, and their cross force relations are aggregated to understand the current situation” (Salerno, 2008). In the same period, Endsley introduced a theory of situation awareness for dynamic systems. In her studies, Endsley focuses on the awareness of the worker, not of the robot

(Wickens, 2008). She established a three-step process to define the situation awareness: Perception, Comprehension, and Projection (Endsley, 1995b). Various methods of measuring human situation awareness are presented in Endsley (1995a). Unfortunately, these techniques, which range from indirect measures such as performance measures to subjective ones such as questionnaires, these techniques are hardly applicable to robots (Dahn et al., 2018). In the period from 2018 to 2022, selecting the first 100 hits of the 481 publications listed in the Web of Science under the query of “title contains situation awareness”, only five contribution papers in the English language are related to the awareness of technical systems. To this end, the authors agree with the finding of Dahn et al. (2018) that many approaches use the term situation awareness without giving a definition. Among the five contribution papers, Burova (2021) argues for the use of small and fast ontologies for fast decision-making to gain situational awareness from ontologies in real-time. The authors take up the idea of a set of small meta-models to measure the quality of the context. D’Aniello et al. (2018) modify Endsley’s scheme for seamless learning. In their attempt to understand the quality of the learned concepts, they describe a metric for the quality of context awareness, which is an important aspect of situation awareness. The authors extend this idea by using an adapted Levenshtein distance instead of simply counting the number of elements. Blasch et al. (2019) discusses information fusion with deep multimodal image fusion according to the JDL scheme and metrics to measure the fusion quality. They argue that different metrics need to be combined to describe situation awareness quality. Yusuf and Baber (2022) apply the distributed situation awareness model to teams of both human and robots. They use Bayesian belief networks under limited information to achieve situation awareness by focusing on perception and projection. They describe a “relevance metric” that measures the accuracy of projection of a subset of agents, and a “transition metric” that measures the quality of a predicted value. However, the metrics of both approaches are specific for the respective deep learning approach and do not apply to cobots.

In the service domain, Sirithunge et al. (2019) proposes an auto-regressive model to recognize the level of interest in interacting with the robot. Focusing on the human–robot interaction, the situation awareness reflects the emotional state of the human working with it. Therefore, they define the user’s level of interest to characterize the situation. The authors take up the idea of including both physical and non-physical aspects. However, the question of how to build an expectation of the human’s intention not the focus of this paper. The work of Dahn et al. (2018) comes closest to this work. They transfer the concept of situation awareness to autonomous agents and propose to measure situation awareness in terms of its opposite, surprise. The authors of this paper pick up this idea in the measurement of consistency

and the measurement of coverage. Moreover, Dahn et al. (2018) follow the same approach to formally define the situation awareness to derive a protocol to improve it. However, from the authors' point of view, context and the situation differ. In conclusion, the context awareness and the situation awareness are different.

Furthermore, the concept of Dahn et al. (2018) builds on aspects, which they define as rules formulated in simple logical expressions that describe the environment. In contrast, the authors use states to describe the environment. This type of modeling allows for more convenient inclusion of uncertainties such as tolerances. Furthermore, the authors disagree with the statement that situation awareness is a binary property. If situation awareness is below 100%, the authors agree that the system may fail surprisingly since the one missing aspect makes the difference. Often, missing a relevant aspect will lead to a non-optimal but usable solution. For this reason, it makes sense to reason about the state of awareness. Moreover, it is easier to improve a continuous quantity than a binary one. Finally, the framework of Dahn et al. (2018) does not say whether a system is situation-aware, but rather that it is not. It is limited to surprise, but does not consider parameters like precision and uncertainties in information processing. The DT and its simulation gap focus as a step towards real-world problems.

*The DT and the Simulation Gap.* Driven by the idea of fully simulatable aerospace missions, NASA started the vision of the DT in 2012 (Glaessgen & Stargel, 2012). The first approach painted the DT equipped with a set of models that cover every detail of the system. However, this approach showed several drawbacks rendering this approach unrealistic or at least uneconomic (West & Blackburn, 2018). Consequently, the survey on the DT (Löcklin et al., 2020) hardly found full-featured DTs. Nevertheless, this field progresses a lot, just with an adapted strategy. Operational simulation becomes one core characteristic of the DT. The synchronization characteristic emphasizes the reality-to-simulation transfer, keeping the cyber-world consistent with the physical asset (Ashtari Talkhestani et al., 2018, 2019). In addition, the author's previous work showed how to bring real-time information, such as data from real-time locating systems, to the simulation and the DT (Ruppert & Abonyi, 2020). More recent work considers intelligence integration into the DT (Jazdi et al., 2021). Waving away the claim of modeling the asset perfectly accurately, the research on the simulation gap (Mouret & Chatzilygeroudis, 2017) comes into touch with the DT research. It also implies that the situation awareness is at stake and cannot be assumed without further measures. Approaches exist that tune the simulator to narrow the simulation-to-reality gap (Collins et al., 2020), but do not yet solve the problem entirely. Following a different approach to bridge the simulation-to-reality gap, Zhao et al. (2020) identifies the key aspects: system identification,

domain randomization, domain adaption, and learning under disturbances. The core difference of the DT compared to the former pure simulation is the direction of the transfer. Instead of transferring a build simulation to reality, the DT runs operational simulations that have to be adapted to the perceived real-world. To this end, Müller et al. (2022) proposes a method to close the reality-to-simulation gap. However, it does not tackle the question of situation awareness and situation consciousness.

In summary, the measurement of consciousness has been applied to humans but has yet to be adapted to cobots. Specifically, the metrics need to be adapted for the cobot application. In the literature, the term "situation awareness" is often used without a precise definition. There are some very specific approaches to measure situational awareness for specific algorithms—more as a benchmark. This type of situational awareness is not suitable for improving collaboration. Furthermore, situation awareness in the literature is limited to simple binary logic expressions. A summary of the related works with the sources (src.), the relevance and the novelty is presented in Table 1. To fill this gap, the authors propose a novel approach to improve the situation awareness of cobots for better human–robot collaboration. Using the concept of perception, comprehension, and projection the cobots learn to adapt like the human awareness process. This way, the authors contribute to a human-centric perspective. By giving a precise definition of situation awareness, the authors generalize the concept of situation awareness to cobot systems away from an algorithm-specific quantity. The concept is extended to a continuous three-dimensional metric that directly measures situation awareness. To improve situation awareness, this work targets the reality-to-simulation transfer. This is the opposite transfer direction compared to most approaches in the literature. Using the situation awareness metric as a guide, the approach automatically determines whether the original model is (in)valid based on the knowledge of the situation. Therefore, the adaptation is much more sample efficient compared to conventional model tuning approaches.

### **Situation-consciousness: the measurement of situation awareness for cobot**

The term awareness is studied for automation systems, e.g. in context-awareness (Kulkarni & Rodd, 2020), situation awareness (Endsley, 1995b, 1996; Rizzi et al., 2017) and risk-awareness (Zhang et al., 2022). Endsley defines situation awareness as: "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" (Endsley, 1995b). It should be noted that Endsley defined the situation awareness with human factors

**Table 1** Summary of the related works

Src.	Relevance	Novelty of this paper
Cohen et al. (2022)	Significant considerations related to cobot acquisition and deployment	Metric to measure understanding of situation
Thorvald et al. (2021), Faccio et al. (2022), Romero and Stahre (2021), Ruppert and Abonyi (2020), and Endsley (1995a, b)	Deep perception, awareness and understanding of human awareness and human factors	Translation to cobot domain
El Ouanjli et al. (2022), Xie et al. (2018), Collins et al. (2020), and Lindemann et al. (2021)	Detection of anomalies and model adaption	Samples reduction, insert situational knowledge
Ashtari Talkhestani et al. (2018), West and Blackburn (2018), and Pairet et al. (2019)	Digital Twin provides model management and synchronization	Add model selection and parameter adaption
Dahn et al. (2018), Endsley (1995a, b), Irvine (2013), White (1991), Wickens (2008), and Salerno (2008)	Measure consciousness and build awareness for humans	Adapt approaches for cobots
Burova (2021)	Context-awareness based on structural models	Extend approach for data processing models
Yusuf and Baber (2022)	Bayesian belief networks under limited information for perception and projection	Add comprehension step
Dahn et al. (2018)	Measure situation-awareness by surprise as a binary quantity	Extend metric to continuous quantity and differentiate the model types

in mind. For this reason, the measurement methods described in Wickens (2008) do not apply.

Nevertheless, this definition itself applies to cobots. Breaking down this definition connects the context of the terms (environment, time and space, meaning) with the situation and the prediction. According to Dey, the context  $C$  is “any information that can be used to characterize the situation of an entity” (Dey, 2001). In this case, the entity is the cobot, and “any information” can be understood as a set of pieces of information. What remains undefined at this point is the term of the situation. The part “can be used” refers to relevance defined by Dahn et al. (2018). According to Salfinger, the situation corresponds “to a particular state of affairs in the observed environment” (Salfinger, 2020). However, instead of talking about several situations at the same time, the authors follow (Salerno, 2008), who sees the situation not only as a state, but rather as a set of states. Further-

more, the authors follow (Wickens, 2008), where the time and the place are also considered important to characterize a situation. However, Dahn et al. (2018) points out that the definitions fall short of several steps by not providing a clear guideline on which to base an implementation. They also do not answer the question of how an agent can achieve context or situation awareness. Therefore, the authors propose formalized definitions for situation awareness and related terms.

### Formalized definitions

In this section the authors puzzle together the core of the definitions from the literature and formalize them mathematically to make them applicable to the cobots. The first important term is context.

*Context.* The context  $C$  depicts a set of objects  $O^k = \{O_1, O_2, \dots, O_N\}$ ,  $N \in \mathbb{N}$  around the cobot and their rele-

vant relations  $R^k = \{R_{ij}^k\}$ , where  $R_{ij}^k$  describes the relation between the  $i$ th and  $j$ th objects at the  $k$ th point in time and  $i, j = 1, \dots, N$ :  $C^k = \langle O^k, R^k \rangle$ .

As Burova (2021) and Yujian and Bo (2007) point out, the context modeling relies on small meta-models to be fast enough for real-time applications. For the same reason, the system focuses only on the relevant objects and relations. Relevance depends on whether a piece of information (object or relation) contributes to current decision making. It is time-dependant since an object may be irrelevant at time  $k$ , but get relevant at  $k + 1$ .

Unfortunately, in real robotic systems, there is no ground truth about the context. Instead, the system must infer the objects  $O^k$  present based on its measurements  $M^k = \{M_1, M_2, \dots, M_K\}$ ,  $K \in \mathbb{N}$  and reason about the relevance. A measured value is an output of a sensor or model at a certain time  $k$ , e.g., a camera image or a laser scan. It is a challenge to extract the context from the measurements. The capability of mastering this challenge is context awareness.

*Context Awareness.* Context awareness is the ability to derive the context from the measurements. Mathematically speaking, context awareness is a function that maps the measurements to the context.

$$f_{ca} : M^k \rightarrow C^k. \quad (1)$$

A proposal for discovering the objects and their relations using small metamodels is described in “The process of improving situational awareness” section. In addition, the objects are related to another important term, the situation. The situation essentially describes the state of the objects.

*Situation.* A situation  $S$  is the set of states of the context objects  $S^k = \{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_N\}$ . Unlike the context, the situation does not refer to a specific entity. Therefore, the system itself is part of the situation. Let  $\mathbf{X}_s$  be the system’s state vector and  $\mathbf{X}_{O,i}$  the state vector of the  $i$ th object. Then the situation  $S$  is the set of the system or object states:  $S^k = \{\mathbf{X}_s, \mathbf{X}_{O,1}, \dots, \mathbf{X}_{O,N}\}$ . The situation is relative to a point in time  $k$ . The relation to the space is relative to the objects in the environment and part of the state vectors. Respectively, situation awareness is defined.

*Situation Awareness.* According to Endsley’s definition, situation awareness includes perception, comprehension, and projection. It is a function that maps from a set of measurements  $M^k$  and the current situation  $S^k$  to the context  $C^k$  and the future situation  $S^{k+h}$ :

$$f_{sa} : \begin{pmatrix} M^k \\ S^k \end{pmatrix} \rightarrow \begin{pmatrix} C^k \\ S^{k+h} \end{pmatrix}. \quad (2)$$

In this equation, the perception maps a set of measurements  $M^k$  to a set of objects  $O^k = \{O_1, O_2, \dots, O_N\}$ . The comprehension connects the objects  $O^k$  with relations  $R^k$ .

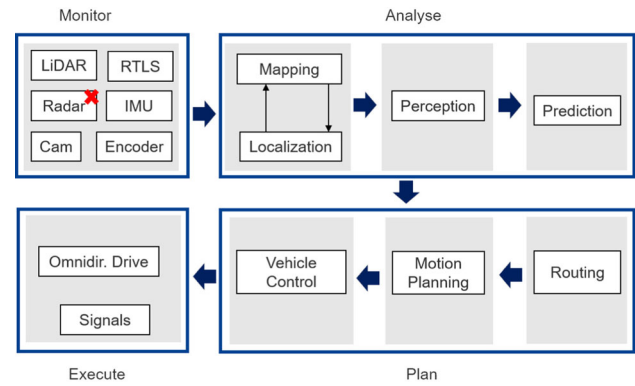


Fig. 1 Architecture of cobot adapted from Zilly (2023)

Perception and comprehension together form the context  $C^k$ . The projection refers to the prediction of the states of the objects in the environment in the near future using the present data. These states of the objects correspond to the definition of the situation  $S^k$ . The near future is modeled with the prediction horizon  $h \in \mathbb{N}_0$ . In conclusion, situation awareness is about building an expectation about the environment and its future state. If a strong deviation to this expectation occurs, this is a disruptive event.

*Event and Disruptive Event.* An event is anything that happens, especially something important or unusual. An event becomes disruptive when it causes the system or environment to deviate strongly from the modeled behavior and is relevant to the system.

### Cobots’ models contributing to situation awareness

Depending on the specific application, the cobots’ models change. Nevertheless, there are some types of models that relate to the basic architecture of cobots. To discuss these models, we first introduce the cobot’s architecture. The architecture is a modified version of the BOSCH automated driving architecture (Zilly, 2023), visualised in Fig. 1.

Automated driving and cobots differ only in their sensors and actuators. This fact is reflected by the cobot’s architecture, a modified version of the BOSCH automated driving architecture (Zilly, 2023), visualised in Fig. 1. Both use SLAM, perception, and prediction to plan routing, motion, and vehicle control, with sensor fusion models processing data in the monitoring step. The SLAM algorithm provides a map, which is enriched with state information of objects in the perception step. The prediction step provides future states of the environment, resulting in an extended map in the form of structural models. Behavioral models contain system goals, routing strategies, and trajectory planning rules, and describe the behavior of other robots and humans using meta-models. However, these models cannot be automatically adapted. Two types of models are subject to the adapta-

tion process: data processing models, which are checked for consistency, and structural models, which are checked for completeness. Reliability and robustness issues affect both types of models, leading to novel metrics for situation awareness proposed in the next section.

### The novel metrics of situation awareness

Broadly speaking, the quality of situation awareness is about rarely being surprised by perturbing events. In addition, the system should be aware of the uncertainties by estimating the size of the gap between reality and simulation. For this purpose, it is less important how the individual deviation will turn out, but rather to be able to predict in which range the spread will lie. Thus, the assessment of situation awareness is characterized by model consistency, context awareness, and model coverage. Together they make up consciousness.

As defined above, context awareness denotes the ability to correctly infer the context  $C^k$  from the measurements  $M^k$ . Consequently, the estimated context  $\hat{C}^k$  should be as close as possible to the actual context  $C^k$ . However, due to inaccuracies in the sensors or the models, a gap in the sense of completeness and correctness may occur between the estimate and the ground truth. Examples of a gap would be a misclassification of an object. The gap  $E_C$  between the estimated contents can be described as

$$E_C = 1 - \frac{\|\hat{C}^k \cap C^k\|_L}{\|\hat{C}^k \cup C^k\|_L}, \quad (3)$$

where  $\hat{C}^k \cap C^k$  represents the subgraphs of the estimated context and the actual context, which are identical, and  $\hat{C}^k \cup C^k$  represents the joint graphs. As a distance metric, the authors propose to adapt the Levenshtein distance (Yujian & Bo, 2007) for graphs, which counts the changes required to transform one graph into the other. To normalize the quality to a value between zero and one, the authors introduce the reference value  $E_{C,ref}$ . This value represents the expected deviations and must be defined by the user's experience.

**Context Awareness Quality ( $Q_{CA}$ ).** The  $Q_{CA}$  measures the similarity of the true context with the estimated one and is modeled as:

$$Q_{CA} = \begin{cases} 1 - \frac{E_C}{E_{C,ref}}, & \text{if } E_C \leq E_{C,ref}, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The way  $Q_{CA}$  is constructed models the uncertainty of the structural models in comparing the two graphs: the assumed context graph and the context graph that combines the possible other context graphs with lower probability. Measuring the similarity of these two graphs articulates the system's confidence in its context perception, which correlates with the model's reliability.

The second quality metric measures consistency. Inconsistencies can occur when different sensors or models infer non-identical states of the situation. A classic example would be redundant sensors that differ in their results or a gap between prediction and measurement.

The degree of consistency is measured by the weighted deviation of each information source from the estimated true value. Following the context awareness quality scheme, the error vector of the system state is  $E_{Con,i} = \|\hat{\mathbf{X}} - \mathbf{X}\|$ , where  $\hat{\mathbf{X}}$  represents the estimated true state and  $\mathbf{X}$  represents the output of a certain measurement or model. Again, the metric is normalized to a reference error vector  $E_{Con,ref}$  determined from experience.

**Degree of Consistency (DoC).** The degree of consistency measures the similarity of all different information sources representing the same quantity of the situation's state vectors. Let  $E_{Con} = (E_{Con,1}, E_{Con,2}, \dots, E_{Con,N})$  be the vector that summarizes all  $i$  discrepancies in the state vectors  $\mathbf{X}_i$ . Then the DoC-related quality is modeled as follows:

$$Q_{DoC} = \begin{cases} 1 - \frac{E_{Con}}{E_{Con,ref}}, & \text{if } E_{Con} \leq E_{Con,ref}, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

The way  $Q_{DoC}$  is constructed models the uncertainty of the data processing models by comparing different sources of information (models or measurements). In this way, the confidence of the system in its predictive capabilities is measured, which correlates with the reliability of the data processing model.

Finally, the quality of coverage the models provide must be mathematically defined. Loosely speaking, the coverage quality represents the certainty of not getting caught by surprise. Surprise is defined in Dahn et al. (2018). Formally, the coverage describes the absence of disruptive events or the ability of the system to model possible scenarios correctly. A scenario  $s_i$  is a sequence of events. Consequently, the model coverage quality can be described by the probability that the system correctly assesses the situation and its state. For this purpose, the measurement of the degree of coverage builds on the previously defined quality metric of consistency.

**Model Coverage Quality ( $Q_{MC}$ ).** The model coverage quality measures the probability that the currently active set of models can accurately model the system behavior. Let  $Q_C$  be the context-modelling quality as defined in Eq. 1,  $Q_{Con}$  be the DoC as defined in Eq. 2. Let further  $s_i$  be a randomly selected, possible scenario. Then the  $Q_{MC}$  is given by

$$Q_{MC} = P(Q_{CA} > 0 \wedge Q_{DoC} > 0 \mid s_i). \quad (6)$$

The proposed approach to determining  $Q_{MC}$  follows the frequentist approach of counting the number of different scenarios between two violations of the criterion  $Q_{CA} >$

$0 \wedge Q_{DoC} > 0$ . The consciousness of a cobot is defined for this purpose.

The way  $Q_{MC}$  is constructed reflects the generalization ability. For each exception that is added, the change in  $Q_{MC}$  tells about the number of cases covered by it. This is how the robustness of the active model set is evaluated.

**Situation Consciousness.** The situation consciousness  $\zeta$  describes the level of situation awareness, i.e. the quality of the function  $f_{sa}$ . Let  $Q_{CA}$  be the context quality as defined in Eq. 1,  $Q_{DoC}$  be the DoC as defined in Eq. 2 and  $Q_{MC}$  be the coverage quality as defined in Eq. 3. Then, the Situation-Consciousness is the tuple  $\zeta = \langle Q_{CA}, Q_{DoC}, Q_{MC} \rangle$ . The situation consciousness represents the level of the ability to model the system and itself with an acceptable reality-to-simulation gap (Müller et al., 2022). The next section details the process developed to improve situation consciousness.

## The process of improving situational awareness

After formally defining consciousness, this section presents a systematic method for improving awareness.

Adapting models always carries the risk of corrupting the model, which can have serious consequences. On the other hand, doing nothing when you know that a model's prediction is deteriorating is also risky. Simply bringing the system to a safe state is not a solution either, because it reduces the reliability of the system too much. As Reich and Trapp (2020) argue, dynamic risk management is needed to dynamically validate these changes as they occur, in order not to lose too much reliability or run into unacceptable risks. To this end, awareness helps as a guiding metric to adapt models only when urgently needed. This concept selectively changes the models to keep the validation effort to a minimum. Another advantage is that fewer samples are needed for adaptation, since they are difficult to obtain, especially in the corner cases.

As discussed earlier, awareness is related to consciousness. According to Endsley (1995b), the awareness of automated systems follows a three-step process: Perception, Comprehension, and Projection. However, this process is focused on the human operator. Table 2 maps the human situation awareness process to cobot situation awareness.

This adapted framework manifests itself in the structure visualized in Fig. 2. The first step in this process is to measure the DoC, which is done in the Perception Step. In this step, the iDT combines and compares the real-world (asset) data and the cyber-world estimates. As a result, the iDT estimates the true state, the error, and thus the  $Q_{DoC}$ . It passes this result to the Comprehension Step. The Comprehension Step combines the context estimation from the virtual world and the context recognition on the real world data to provide

**Table 2** Adapted situation awareness for mobile robots

Steps	Interpretation for cobot
Perception	Perceive deviations from forecast
	Perceive deviations between models
	Perceive disruptive events
Comprehension	Anomaly detection
	Retrieve the context around the system
	Characterize disruptive events
Projection	Synchronize model and asset
	Predict future situation

further quality estimates, namely the context quality and the coverage quality. From this understanding, the iDT concludes the Projection Step. In this step, a machine learning model generates a correction model. This correction model is later tested on collected real-world data to validate its generalization to previously observed situations. In the virtual world, the iDT predicts the situation  $S^{k+h}$ , which serves as a witness to validate the quality of the updated model.

In this way, the knowledge grows with the experience the iDT gains during operation, adapting the DT to its deployed environment. The following subsections describe this situation awareness process in detail.

### Perception

The uses the perception step to observe itself interacting with the environment (Fig. 3). The key to this step is to build an expectation of the situation  $\hat{S}$ , i.e. the state vectors  $\hat{X}_i$ , and compare it to the available information. However, as Mouret and Chatzilygeroudis (2017) shows, synthetic data from simulators differ significantly from real process data. In general, models simplify reality and therefore must first be made comparable by design. On the other hand, real-world data must first be cleaned to reduce the complexity of the relevant aspects. To this end, the iDT performs data acquisition, pre-processing and transfer steps. It distinguishes between two domains: the cyber and the physical world.

In the cyber domain, the simulation environment produces synthetic data. Typically, this data represents a subset of the total space of possibilities in which the system operates. It is very specific to the case being simulated. To make the data more general, noise and contamination effects can be added. Moreover, the iDT extends the covered exploiting domain randomization to prepare the system for real world data. Concrete approaches to how this works are proposed in Tobin et al. (2017). The result of this domain randomization is synthetic features that need to be unified to match the process features. It should be noted that the algorithms in the simulation domain may differ from those in physical space.

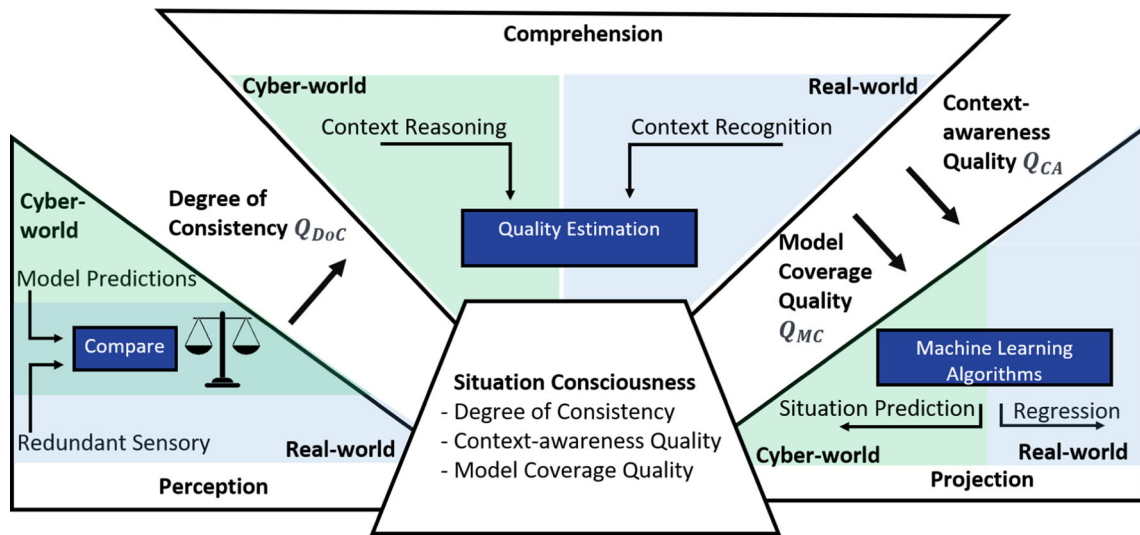


Fig. 2 Situation awareness scheme building consciousness

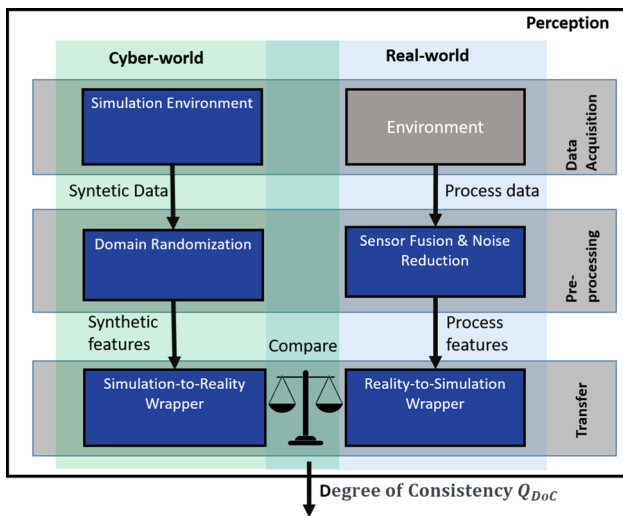


Fig. 3 The perception step

The Simulation-to-Reality Wrapper takes care of this task. It puts the detection layer on a higher level and facilitates the comparison. An example is the object recognition domain. In this domain, the identified label and the confusion matrix to other labels should be considered. However, reducing this matrix to the 5–10 most relevant misclassifications and comparing them between the cyber world and reality is more feasible than comparing the features used by the object detectors. In the physical world, the sensor captures the process data of the object. Since real-world data problems are the opposite of synthetic data, DT aims to cleanse the data.

To do this, the DT uses sensor fusion and noise reduction techniques to pre-process the data. For example, the multiple measurements from a LiDAR sensor suggest different positions in space. Using Kalman filter-based simultaneous localization and mapping (SLAM), the DT merges these val-

ues into a single feature: the most likely position on the map. Based on the process features, the Reality-to-Simulation Wrapper further abstracts away the channel-specific aspects. Using LiDAR as an example, the position information is provided relative to the robot’s position. Suppose the map predicts where to find a moving object to be compared. In this case, the Reality-to-Simulation Wrapper has to transform the detected objects to the map to compare the simulated position of the object with the real one.

By performing these three steps, estimating the actual value with the value available in the model or sensor can be applied to models in general, as long as a set of comparable features can be extracted. In the case of physical models, physical state variables such as position, velocity, and orientation can be compared. In the case of contextual models, for example, degrees of membership can be used. In addition, the perception can be extended by external feedback, e.g. from a worker. This corresponds to an extension of the vector  $K_j$  or  $S_{r,j}$ . Having perceived the environment, the next step is to make sense of these perceptions. Since the monitoring step makes the different models comparable in the state vectors  $X_j$  and  $S_{r,j}^k$ , the DT computes  $Q_{DoC}$ . This step is described in the next subsection.

### Comprehension

The comprehension step analyzes the statistical properties of the perceptions, classifies the context, and analyzes the coverage quality of the iDT. Figure 4 visualizes this step. The process starts to distinguish the normal ( $Q_{DoC} > 0$ ) and the abnormal data ( $Q_{DoC} = 0$ ) using anomaly detection. In this paper, the anomaly detection is limited to the DoC for the next step. Since  $Q_{DoC} = 0$  already defines that the models

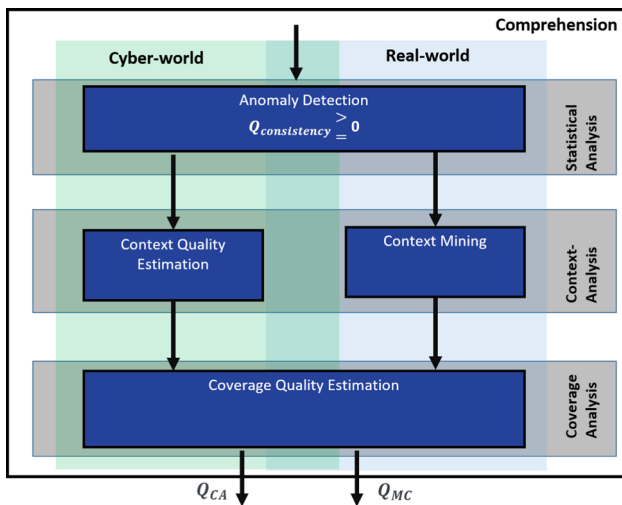


Fig. 4 The comprehension step

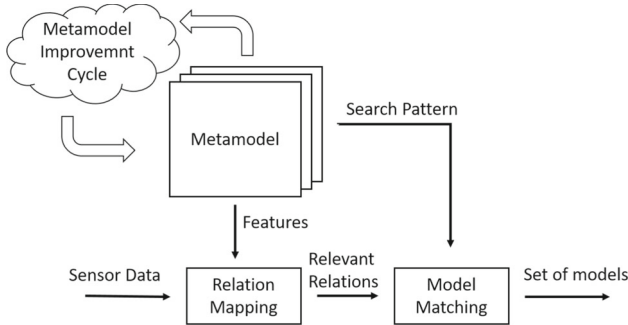


Fig. 5 Context recognition

are inadequate to describe the scenario, this quantity is a natural measure of anomalies. Lindemann et al. (2021) provide a comprehensive survey for an exhaustive view of anomaly detection. In a sense, the sophisticated methods allow to compare different time steps in parallel to identify more types of anomalies. After analyzing the data for anomalies, context comes into focus. Whether an anomaly is detected or not, the iDT relies on either the virtual or physical world. Because an anomaly indicates that the models (i.e., the virtual world) are not sufficiently accurate, the iDT relies on real-world data to perform context recognition.

Figure 5 visualizes the process of context recognition. Context recognition is based on a set of small metamodels as proposed by Burova (2021) and Yujian and Bo (2007). From the metamodel, the relational mapping gets information about what to look for in the sensor data. Examples of contextual information are links between obstacles, as table feet are linked to their table top. The subsequent Model Matching takes the relations and search patterns from the metamodels to check which metamodel to instantiate to create a set of models. In this way, relation mapping determines the relationships between the recognized objects. The cycle

of improving the metamodel is excluded from the scope of this work and will be left for future work.

Using the context recognition described above, the iDT estimates the impact of disruptive events. However, this method can only detect correlations, not causality. Another instance must validate the observation to cover causality.

In the case of normal data, the available models prove to be usable. Since the predictions fit, the estimated context is close to the estimated one. The iDT analyzes the frequently occurring patterns associated with a specific context. Furthermore, the iDT estimates the probability of certain objects or conditions (represented in situation  $S$ ) occurring in a given context. In this way, the estimate of the ground truth situation  $S$  is complemented by elements with a very high probability of being present in the current situation. Similarly, discarded objects close to the detection threshold can be considered in  $S$ . In addition, engineering knowledge flows into the estimated situation and the assumed ground truth. For example, the table top is hard to detect for a 2D laser scanner. However, if the legs of the table are a certain distance apart, the robot can infer that this should be a table. From this data, the current context reveals that there is another object, a tabletop, which is invisible to the senses. After analyzing the context in this way, the system has both variables available and therefore calculates the context quality. In the event of an anomaly, the DT relies on real-world data to rebuild the context model. Based on the context model, the situation is identified. This is part of context mining.

**Situation Identification.** Based on the context model, the situation is identified. For this reason, the current state of the objects is estimated. In this respect, both pattern recognition and regression algorithms are used. When an anomaly is detected ( $Q_{DoC} = 0$ ), the anomaly must be categorized by searching for similar situations with clustering algorithms. A distance criterion is certainly available for this purpose. The consistency error  $E_{Con}$  is a rough distance metric, where a similarly high error is expected for similar situations. An example of anomaly categorization is published in Müller et al. (2022). Independent of whether there is an anomaly or not patterns of the context may be used to identify similar situations. As a suggestion, the graph similarity of the context model might serve as a further feature and based on the metamodels even more features might be extracted. The detailed similarity metric and features to identify the situation are a design choice. The authors decided to attach these features and the parameters to estimate in order to form the object's state to the object properties. Having an increasing data set of context features associated with object's state, the identification of the set of the object's states, i.e. the situation identification increases in precision. However, more research has to be conducted in order to come up with automated feature extraction and labelling for situation identification. Given the labelled data set, clustering identifies different

modes of the object and regression algorithms estimate the current state and may predict future ones.

In the subsequent coverage analysis, the system checks whether a similar case has already occurred. The coverage quality estimate changes only if the current situation is new to the system. For abnormal data, the system cannot measure or predict the environment correctly, which causes the coverage quality estimate to decrease. For normal data, the coverage quality increases.

Based on these analyses, the iDT predicts the future development of the situation. As a result, the comprehension step provides context and coverage quality. This step is described in the next section.

## Projection

The final step of the situation awareness scheme is projection. In this step (see Fig. 6), the future situation  $S^{k+h}$  is predicted based on an updated version of the models. To update the model, the iDT creates a data-driven correction model that depends on the context. For this purpose, the Data Selector selects the samples from the real data based on their similarity to the current context. The context error (Eq. 3) serves as a distance metric. Based on this training data, a machine learning algorithm creates a mapping that takes the output of the original model and modifies it to get closer to the actual value. Machine learning algorithms, especially regression algorithms, provide a correction model for both regression testing and predicting the future situation. Disturbing events limit the space in which the correction model is valid. Aniculaesei et al. (2018) call this scenario space a dependability cage. As the sample size increases, the agents learn how to compensate for the event characteristics and the error between predicted and actual states. A new correction model emerges. Once the model converges, the additional samples for the same context expand the dependability cages.

With increasing samples, the iDT gains confidence in the context in which the updated models are appropriate. Typically, the updated models are run in parallel with the existing models in a test phase. In this test phase, regression tests are performed. Once the iDT is confident in the quality of the model, it releases the sandboxed models and puts them into production mode. In case of unintended behavior, the model contains a link to the old model as a fallback plan. After the models have been updated and approved, the prediction of the future situation takes place. To do this, the iDT first synchronizes the models with the asset. During synchronization, all situation states in the models are set to their most likely values. These updated states are fed into the prediction module. Based on the updated states and models, the prediction module predicts the future situation  $S^{k+h}$ .

Finally, the iDT communicates the system's state awareness to the user. This step is described in the next section.

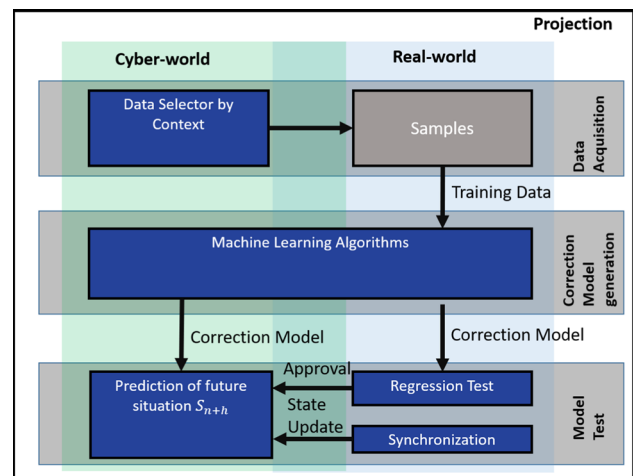


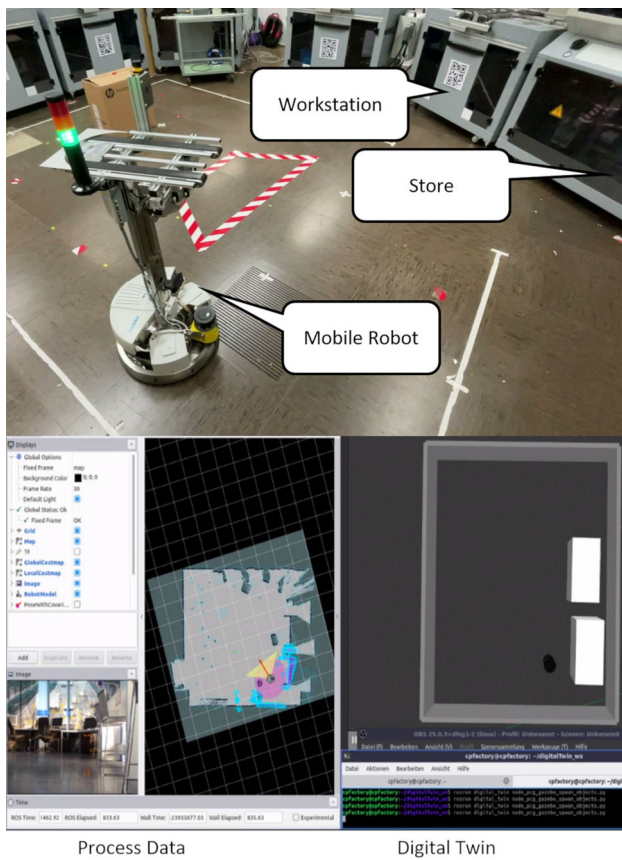
Fig. 6 The projection step

## Communicating awareness using intelligent Digital Twin

The communication of situational awareness includes the visualization of the models. The iDT shows the running simulation processes based on real-time data exchange. For example, the map and the simulated movement of the cobot are shown together with incoming camera images in real time. Tablets can be used as a front to display the visualization. This allows the worker to inspect and analyze the cobot's awareness. If less detail is required, a dashboard displays the previously introduced metrics of coverage quality and DoC, and allows for the review of specific consistency errors.

## Experiments and results

To study situational consciousness, a cyber-physical model factory with a mobile robot platform of the type Robotino 3 Premium from Festo, an automated warehouse, and four workstations is considered. The Robotino serves as a collaborative robot and is focused on this work. The mobile robot uses a laser scanner for SLAM. Specifically, the Adaptive Monte Carlo Localization (AMCL) algorithm is used. A monocular camera supports object recognition and visualizes the environment. For the communication interface, a Robot Operating System (ROS) node runs on the robot, providing a service-oriented interface. A PC running Ubuntu 20.04 wirelessly controls the Robotino through this ROS node. It mimics a cloud server and is equipped with an i9 processor and NVIDIA P620 graphics card. The simulation environment is based on Gazebo 10. It is complemented by grid maps generated by the laser scanner and machine learning



**Fig. 7** Visualization of the cobot and its iDT

models. The visualization is done with the Rviz framework and the algorithms are implemented in C/C++ and Python 3.

Figure 7 illustrates Robtino within the cyber–physical factory and its iDT. An iDT is set up for the Robotino. The robot’s iDT manages safe navigation in the factory, but also an office environment using a probabilistic roadmap planner as a global planner. The robot works closely with human workers, providing tools and workpieces. When working closely with robots, the worker must understand the robot’s situational consciousness. For this purpose, the authors take the continuous example of the correct movement and positioning of the robot in simulation and reality. This example is deliberately chosen to illustrate the idea. More complex examples, namely an object recognition ontology and a human motion model, complement the continuous example.

This scenario is intended to show the evolution of the robot’s consciousness. During its journey, the robot perceives objects such as workstations and elements of the office environment such as tables. The experiments follow the structure of the previous section to evaluate the process of developing consciousness and its communication with humans.

An important task of the iDT is to communicate the mobile robot’s knowledge about its environment. In particular, the human operator needs to know when the system perceives

the environment differently than the operator would expect. In this context, the visualization of the iDT is crucial. Figure 7 (left-bottom) illustrates the perception of the iDT. The laser scanner detects I-shaped tables. The robot maps it to a table by pattern recognition, which the iDT represents as bounding boxes that the robot is not allowed to enter. The robot’s perception is structured in several views, which the worker selects in the toolbar on the left. The map shows the current laser scanner measurement, the direction of movement, and the associated grid maps that mark the forbidden areas. The terminal on the bottom right gives feedback about the running scripts.

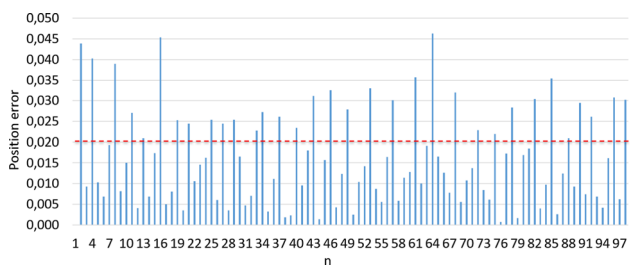
In this section, the results of the experiments are discussed according to the previously detailed process steps. The calculation of the DoC for positioning is described in “Perception” section and the comprehension step for anomaly detection in “Comprehension” section. Finally, the results of the projection step are described in “Projection” section.

## Perception

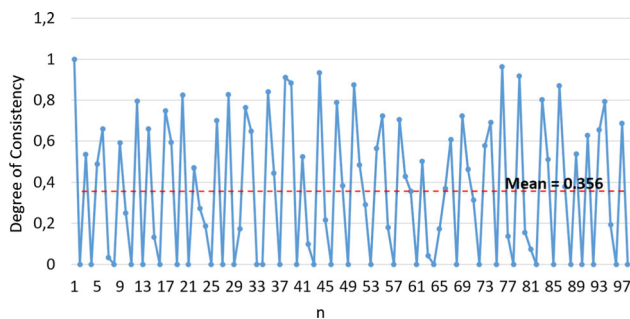
The perception step shows the calculation of the DoC for the positioning. It also evaluates the position of obstacles on the map. Besides the obstacles, the position of the human worker has to be estimated. The experiment is limited to these three aspects to keep it simple. Considering the close cooperation with the human worker, the reference error  $E_{Con,ref}$  is set to a maximum gap between simulation and reality of 2 cm. The indoor positioning system tracks human motion. To reflect the inherent position uncertainty, the reference error  $E_{Con,ref}$  is set to 75 cm.

In the simulation, the robot’s position computation uses a simple physics model, assuming that the velocity controller is always on and therefore  $s(t) = s_0 + v(t) \cdot t$ . However, there are two nonlinearities in the motion of an autonomous mobile robot that the models do not take into account. This leads to poor model quality, which is detected and quantified in this step. As a reference, the laser scanner evaluates the position from several data points referenced to known objects. By transforming the SLAM position into the simulation coordinates, the simulated and measured positions become comparable. Figure 8 shows the results without compensation. On average, every third position value exceeds the reference error  $E_{Con,ref}$  (marked in red). Whenever this happens, the system resets the simulated position to the measured value, resulting in frequent synchronizations. Although the synchronizations have been so frequent, deviations above  $2 \cdot E_{Con,ref}$  occur. This causes the autonomous mobile robot to “jump” in the simulation environment. As expected, the initial model quality is rather poor. Figure 9 shows the calculated  $Q_{DoC}$ .

Similar to the data processing model of the robot’s movement, the human’s movement is estimated. We consider a



**Fig. 8** Deviation of the position before compensation and error reference



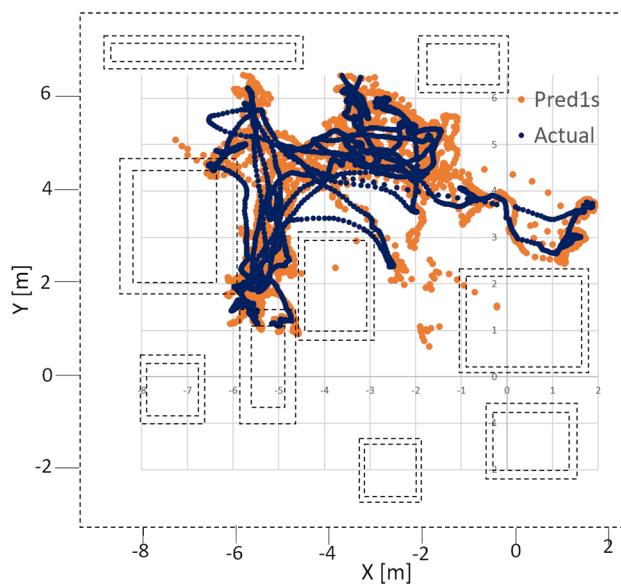
**Fig. 9** Degree of Consistency before the compensation.  $Mean = 35.5\%$

movement pattern of a human searching for something, e.g. a specific document, etc. For this purpose, the speed and direction of the human’s movement is low-pass filtered and based on this information, the next position is estimated. The model assumes that the human will most likely continue with his speed and direction. This assumption is often violated, especially with respect to direction, resulting in significant deviations. The prediction result versus the actual position is visualized in Fig. 10. Correspondingly, the estimation for a time horizon of 3 s has a consistency quality as visualized in Fig. 11 with an average of 32.3%.

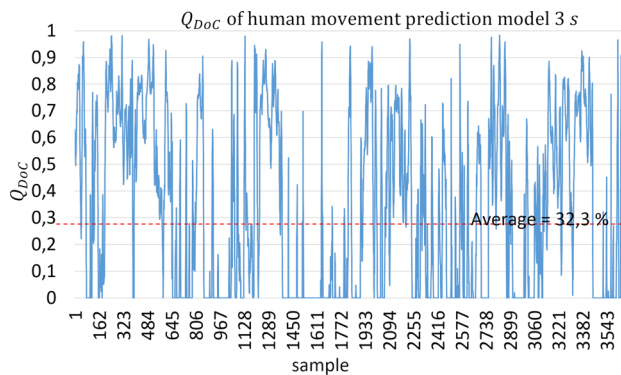
**Comprehension**

The comprehension step begins with anomaly detection. An anomaly is detected whenever  $Q_{DoC} = 0$ . As visualized in Fig. 9, this happens 34 times in the uncompensated dataset.

For this reason, the authors extend the example with object detection and context mining. The map in Fig. 12 shows the result as provided by the SLAM algorithm. It is annotated with the context models associated with each object. Due to the simplicity of the models, they are not hard-coded into the system. Instead, the appropriate models are loaded based on a mapping table. As visualized in Fig. 12 (right), object detection identifies additional closed regions based on the mapping table in Fig. 13. This mapping table uses the context of the detected objects to each other for object identification and representation in the grid map.



**Fig. 10** Human movement model with horizon 1 s



**Fig. 11** Human Model.  $Q_{DoC}$  for prediction horizon 3 s

As Fig. 13 shows, the laser scanner perceives the table legs as a single object at a certain distance (relation). In this way, the detection is invariant to rotation. Furthermore, this representation allows the calculation of the context quality. Things are simple as long as all feet are correctly detected. However, if only three of the four feet are recognized, the table will not make it into the estimated context  $\hat{C}^k$ , but into the reference context  $C^k$ . In the red circle, the algorithm does not estimate a table, but there is a high probability that it could be a table. Only three out of four feet are correctly recognized. They are tables, but they blend into the background.

In summary, the object detection correctly detects four out of six objects (dashed green). For the two tables, one table with food is missing. As a consequence, using the Levenshtein distance, 2 close To-relationships and 1 O-shape object must be added. In addition, the detector misses that the objects form a table of shape V-shape. There are 2 objects and 4 relations missing for each missed table. There are 19 objects and 36 relations, where 16 objects and 28 relations

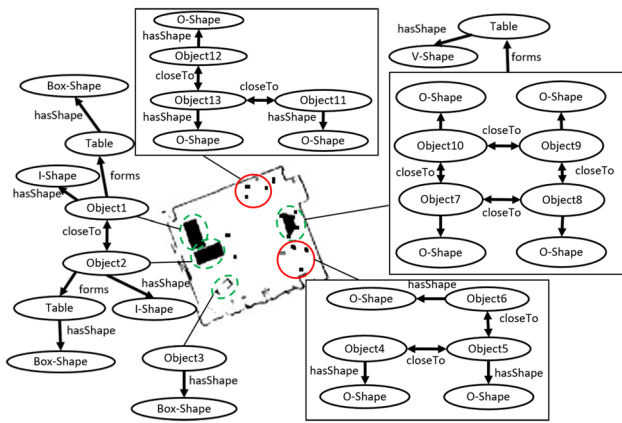


Fig. 12 Object recognition based on context models

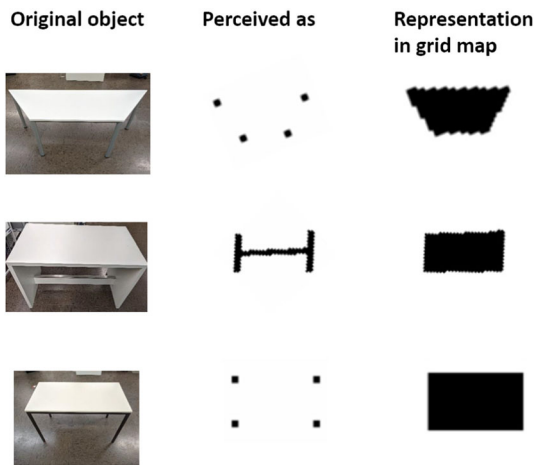


Fig. 13 Object detection rule table

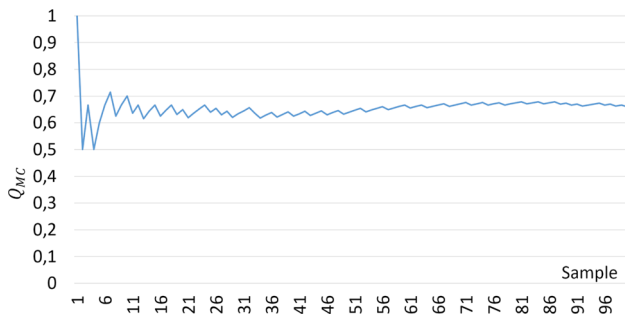


Fig. 14 Estimated coverage quality over samples without adaptation. Mean = 66%

are correctly detected and do not need to be changed. For simplicity, each operation is counted equally (i.e., add-relation or add-object). The context error is therefore calculated according to (3):  $E_{Con} = 1 - \frac{(16+28)}{(19+36)} = 0.20$ . The reference is  $E_{Con,ref} = 1$ , which results in a context quality of  $Q_{CA} = 1 - 0.20 = 80.0\%$ .

If the data is classified as abnormal, data mining is performed on real data. For example, using the camera as an

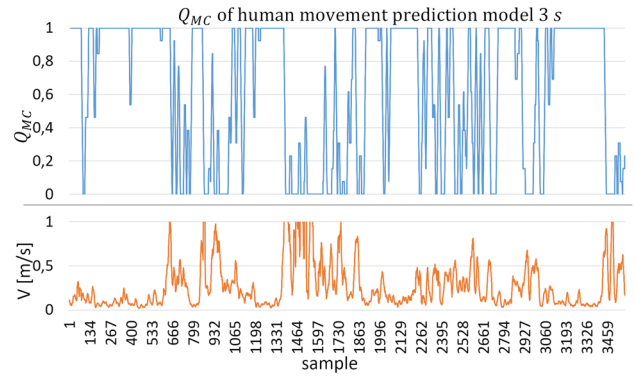


Fig. 15 Estimated coverage quality and velocity over samples for human model

additional source of information, the previously misclassified tables (marked in red) become observable. With this information, the algorithm learns that multiple tables in a row and tables too close to the wall characterize the misclassifications. This shows the limitations of laser scanner based object recognition. In this research area, Sahlab et al. (2021) proposes more sophisticated methods to extract contextual features.

The next step is to compute the coverage quality. As a naive approach, each scenario is re-classified, resulting in a coverage quality as shown in Fig. 14. The estimated coverage quality is simply calculated as the ratio of covered cases to the total number of cases. The coverage estimate converges to  $Q_{MC} = 66.0\%$ .

As for the mobile robot motion model, the model coverage quality is also calculated for the human motion model. In contrast to the overall estimation, we are now interested in a limited history of 10 samples. The corresponding diagram is shown in Fig. 15. The velocity plot below shows that  $Q_{MC}$  is inversely correlated with velocity. Values above  $0.45 \text{ m s}^{-1}$  are associated with an anomaly.

In summary, the direction of human motion is difficult to predict. However, in a stochastic sense, the position of the human can be assigned to a certain area with a well predictable probability. This is the purpose of the coverage quality metric. The cobot adapts its path planning strategy accordingly.

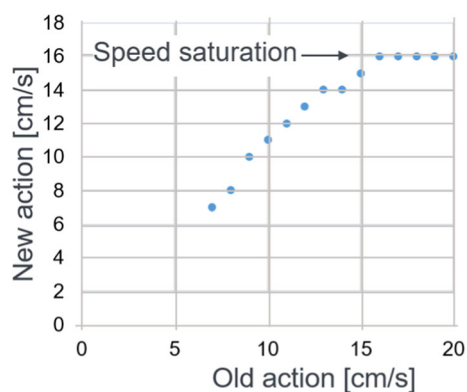
### Projection

The projection step starts with the collection of samples associated with the respective context. This training data is used to train a machine learning regression. The results of the authors' algorithm are compared with the classical offline reinforcement learning algorithm. The parameters of this reinforcement learning algorithm are given in Table 3.

The reinforcement learning algorithm compares the position value of the model before and after synchronization. The

**Table 3** Baseline Reinforcement Learning Algorithm

Parameter	Value
Algorithm class	State–Action–Reward–State–Action
Used Training Samples	8327
Available input	Continuous value Delayed reward Multi-action
Assumptions	Only longitudinal control
Action space	$[0 \text{ cm s}^{-1}:1 \text{ cm s}^{-1}:20 \text{ cm s}^{-1}]$
Reward	$R(x) = \begin{cases} +1 & \text{if } x < 1 \text{ cm} \\ -5, & \text{otherwise} \end{cases}$
End of an episode	$x > 2 \text{ cm}$ or Rotational deviation $> 0.01 \text{ rad}$

**Fig. 16** Mapping of old velocity to actual one using reinforcement learning

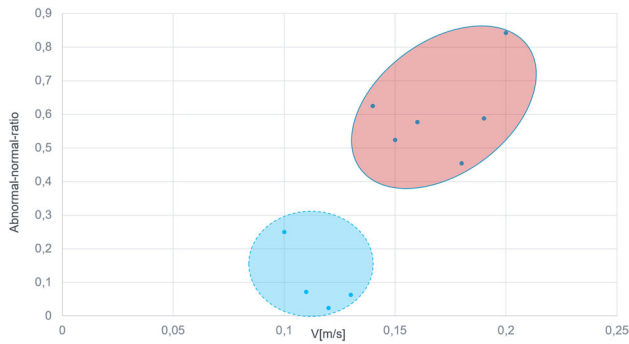
improved model is not applied directly, but runs in parallel with the original model until it is considered stable. As a result, the reinforcement learning algorithm comes up with a mapping table that maps the original velocity (“old action”) to the better fitting velocity (“new action”). The mapping from old to new action is visualized in Fig. 16.

This experiment shows that the resulting model is too simple. In addition, the graph shows velocity saturation at  $16 \text{ cm s}^{-1}$ . The physical controller seems to have a limit at this value instead of the assumed  $20 \text{ cm s}^{-1}$ . This behavior could be a friction that is using up the control reserves. The nonlinearity is validated at the physical plant. As a result, the initial situation awareness  $\zeta_{unadapted} = \langle Q_{CA}, Q_{DoC}, Q_{MC} \rangle = \langle 80.0\%, 35.5\%, 66.0\% \rangle$  is rather low. Applying this compensation shows a significant improvement in the model. The number of deviations was reduced from 33 to 10, about a third compared to the old model, and the magnitude of the deviation was also reduced. Therefore, using the reinforcement learning agent, the situation awareness increased to  $\zeta_{benchmark} = \langle 80.0\%, 42.3\%, 91.3\% \rangle$ .

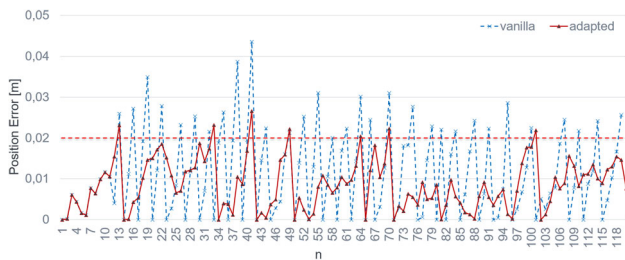
This confirms the validity of the metrics, as the improvement of the models is as expected.

Unlike the robot motion model, the human motion model cannot be further optimized. This is due to the fact that there is no correlation between the previous movement vector and the future one. While the amount of movement is quite predictable—leading to  $Q_{MC}$  in Fig. 15—the direction is completely random. Since no other information is available, the adaptation will not improve the results. Therefore, the human motion model is sent to a human to improve the model. To do this, the algorithm exposes the limitations of the model itself. Nevertheless, the model still provides a benefit, since  $Q_{DoC}$  can be interpreted as a circular area around the estimated position, where  $Q_{MC}$  gives the certainty of the human being within that area. Exemplarily, in the range of  $k \in [292, 583]$ , the estimated probability of the human being within the radius of  $0.75 \text{ cm}$  around the estimate is between 60 and 100%.

However, with respect to the mobile robot’s motion model, the optimization succeeds efficiently. In contrast to the state-of-the-art algorithm, the authors provide the algorithm described in “The process of improving situational awareness” section. Unlike the reinforcement learning agent, which requires 8327 training samples, the proposed algorithm uses only 500 samples. Therefore, the training samples are different. However, since the samples come from the same process, this influence should be rather small. The analysis step, specifically the context analysis and anomaly detection, provides samples to the regression algorithm. The synchronization process takes the measurements from the laser scanner and processes them into position information using the SLAM algorithm. The synchronization module then compares this position with the simulated position. For each target value, the error between simulation and reality is calculated and the ratio between normal ( $Q_{DoC} > 0$ ) and abnormal data ( $Q_{DoC} = 0$ ) is calculated. The ratio is then



**Fig. 17** Clusters of abnormal–normal ratio

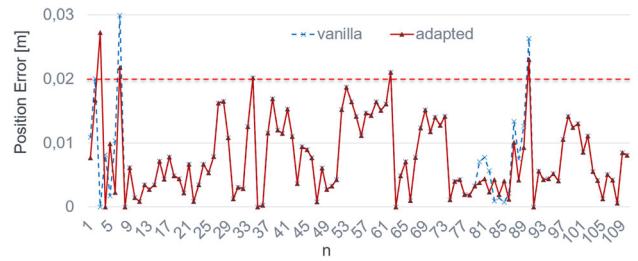


**Fig. 18** Position Error before (blue) and after (red) adaption for  $v > 16 \text{ cm s}^{-1}$  (Color figure online)

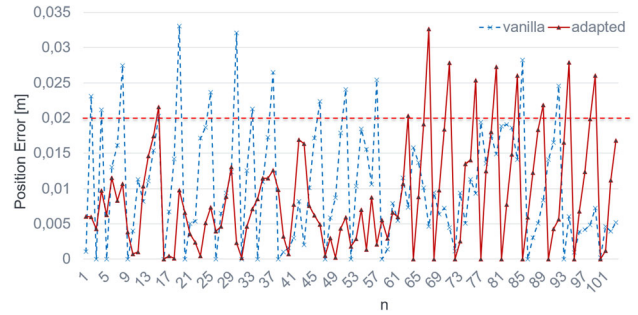
clustered as shown in Fig. 17. The dashed blue cluster contains the samples with a low proportion of outliers, which means that the underlying model is basically correct. The red cluster contains the samples with a high proportion of outliers, which means that the model needs to be improved.

From this clustering, the system learns that the values  $< 13 \text{ cm s}^{-1}$  match quite well, while the values above have a significant error. This results in two regions: the linear region  $[6 \dots 16]$  and the saturation region  $> 16 \text{ cm s}^{-1}$ . The velocity values of  $14 \text{ cm s}^{-1}$  and  $15 \text{ cm s}^{-1}$  show unusual behavior, which will be discussed later in this section. Therefore, no changes will be made to the linear part, while the regression algorithm will try to reduce the introduced error for the other parts. Figure 18 shows the result of the vanilla simulation (blue dashed) and the adjusted simulation (red) in comparison. In this range the consistency quality increases to 58.8%. As shown in Fig. 19, there are no significant changes in the linear region, as the model already works well for this region. The position error for the velocity values of  $14 \text{ cm s}^{-1}$  and  $15 \text{ cm s}^{-1}$  is visualized in Fig. 20. As can be seen from the graph, it works partially with a very low error rate, but in some cases (right part) the error rate increases even higher than in the original version. This leads to an increase of the consistency quality to only 51.8%. Taking all cases together, the results improve to  $Q_{DoC} = 59.0\%$ .

As with the reinforcement learning agent, the quality of coverage increases even more. Obviously, the benefit is highest in the saturation region. The quality of coverage



**Fig. 19** Position error before (blue) and after (red) adjustment for  $v < 13 \text{ cm s}^{-1}$  (Color figure online)



**Fig. 20** Position error before (blue) and after (red) adjustment for exceptions (Color figure online)

increases to 93.2%. In the case of the exceptions, the coverage quality increases to 91.3%. Consequently, the coverage quality in the updated scenario reaches  $Q_{MC} = 93.2\%$ . If we compare the quality metrics before and after the situation awareness process, we see a clear improvement. The situation awareness process improves the situation awareness to  $\zeta_{adapted} = \langle 80.0\%, 59.0\%, 93.2\% \rangle$ . At this point, there is no change in context quality, since the context detection reference value does not trigger an adaptation process. The authors leave the improvement of the context quality for future work. In conclusion, the proposed method shows similar and slightly better performance. The difference in consistency quality varies the most, since the new algorithm does not use quantization. Since the regression is not bound to a specific quantization, it is better in this example. However, the main strength of the proposed approach is not mainly the accuracy, but the required number of samples. In contrast to the RL agent, the approach consumes only about 500 samples.

**Discussion**

The proposed metric can be used in the iDT to communicate situation awareness. The iDT handles the simulations and real-time communication through the sensors and visualization devices. With the management board, these metrics provide a high-level indicator of the cobots’ understanding of the current situation. If necessary, the iDT allows more

**Table 4** Summary of results without (baseline) and with improvement cycle

Metric	Baseline (%)	Our approach (%)
$Q_{CA}$	80.0	80.0
$Q_{DoC}$	35.5	59.0
$Q_{MC}$	66.0	93.2

**Table 5** Summary of results of model improvement with traditional reinforcement learning (baseline) and with our approach

Metric	Baseline	Our approach
Improvement of $Q_{MC}$	25.3%	27.2%
Training Samples	8327	500
Required re-validation of model	100%	63.6%

detailed analysis in real time, checking the models or signals in question. In this way, the approach enables more efficient communication with the human operator. Following the three-step process of situation awareness, a scheme for improving situation awareness was proposed and evaluated using the example of the positioning of the mobile robot platform Robotino. The experiment shows that the quality metric is applicable to the robotic system and qualitatively represents the state of situation awareness. Furthermore, the situation awareness improvement process increased the  $Q_{DoC}$  from 35.5% to 59.0% and the coverage quality ( $Q_{MC}$ ) from 66.0% to 93.2% as shown in Table 4. In summary, the system covered 25% more cases than before, while reducing the gap between reality and simulation. Moreover, as shown in Table 5, the improvement is achieved with way less training samples and less re-validation effort. The improvement process for the context quality is left for future work.

## Conclusions and future work

Situational consciousness can be measured by situation consciousness. Situation consciousness is a quality metric that includes three components: Context Quality, Degree of Consistency, and Coverage Quality. Together, the tuple describes the state of situation awareness. The degree of situation awareness is crucial for smooth cooperation between humans and machines. Only if the human understands how well the robot is aware of its environment, can the human worker adapt her behavior appropriately, such as moving in the other direction, preparing for the approach, or moving away from the robot. This paper introduces the measurement of situation awareness for the domain of collaborative robots based on the iDT.

Speaking of future work, situation identification may be generalized by automatically learn the states of the environment's objects. Moreover, human emotions can be integrated

into situation awareness considerations. Kansei Robotics can help adapt robots to human-centered manufacturing. The Kansei factor could effectively maintain a comfortable state thanks to the emotional synchronization in human–robot interaction (Hashimoto, 2006). It could enrich the context with a non-physical state of the human-model object. Designing cobots with a human-centered approach, taking into account the unique characteristics of machines and technology, can help to further improve the situation awareness of cobots and enhance communication and understanding between humans and machines. To this end, a more efficient metamodel development cycle is enabled, which could be another research direction to investigate. Furthermore, the knowledge of situation awareness allows to improve the cobot's decisions based on the situation and the environment. To this end, risk estimation of the robot's behavior can be performed in real time, taking into account the measured uncertainty that the cobot is currently dealing with. Moreover, collaborative task-sharing research benefits from formalized and improved situation awareness. It is worth investigating how the proposed approach can iteratively increase the recognition of safety-relevant predictions. This approach can also improve communication between machines and humans, leading to greater safety and more efficient collaboration.

**Acknowledgements** Dr. Tamás Ruppert was supported by the “Magyar Állami Eötvös Ösztöndíj”, financed by the Hungarian state.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material

is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Aniculaesei, A., Grieser, J., Rausch, A., Rehfeldt, K., & Warnecke, T. (2018). Toward a holistic software systems engineering approach for dependable autonomous systems. In *2018 IEEE/ACM 1st international workshop on software engineering for AI in autonomous systems (SEFAIAS)*, 2018 (pp. 23–30). IEEE.
- Ashtari Talkhestani, B., Jazdi, N., Schlögl, W., & Weyrich, M. (2018). A concept in synchronization of virtual production system with real factory based on anchor-point method. *Procedia CIRP*, 67, 13–17. <https://doi.org/10.1016/j.procir.2017.12.168>
- Ashtari Talkhestani, B., Jung, T., Lindemann, B., Sahlab, N., Jazdi, N., & Schloegl, W. & Weyrich, M. (2019). An architecture of an intelligent digital twin in a cyber-physical production system. *At - Automatisierungstechnik*, 67(9), 762–782. <https://doi.org/10.1515/auto-2019-0039>
- Azni Jafar, F., Abdullah, N., Blar, N., Muhammad, M. N., & Kassim, A. M. (2014). Analysis of human emotion state in collaboration with robot. *Applied Mechanics and Materials*, 465, 682–687.
- Blasch, E., Liu, Z., Zheng, Y., Majumder, U., Aved, A., & Zulch, P. (2019). Multisource deep learning for situation awareness. In *Automatic target recognition XXIX*, 2019 (pp. 131–142). SPIE.
- Breque, M., De Nul, L., Petridis, A., & Directorate-General for Research, and I.E. Commission. (2021). 01. Industry 5.0: Towards a sustainable, human-centric and resilient European industry. *Research and Innovation Policy*, 1, 25–29.
- Burova, Y. (2021). Knowledge based situation awareness process based on ontologies. In *CEUR workshop proceedings*, 2021.
- Cohen, Y., Shoval, S., Faccio, M., & Minto, R. (2022). Deploying cobots in collaborative systems: Major considerations and productivity analysis. *International Journal of Production Research*, 60(6), 1815–1831.
- Collins, J., Brown, R., Leitner, J., & Howard, D. (2020). Traversing the reality gap via simulator tuning.
- Dahn, N., Fuchs, S., & Gross, H. M. (2018). Situation awareness for autonomous agents. In *2018 27th IEEE international symposium on robot and human interactive communication (RO-MAN)*, 2018. IEEE.
- D'Aniello, G., Gaeta, A., Gaeta, M., & Tomasiello, S. (2018). Self-regulated learning with approximate reasoning and situation awareness. *Journal of Ambient Intelligence and Humanized Computing*, 9(1), 151–164. <https://doi.org/10.1007/s12652-016-0423-y>
- Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), 4–7.
- El Ouanjli, N., Mahfoud, S., Bhaskar, M., El Daoudi, S., Derouich, A., & El Mahfoud, M. (2022). A new intelligent adaptation mechanism of MRAS based on a genetic algorithm applied to speed sensorless direct torque control for induction motor. *International Journal of Dynamics and Control*, 10, 1–16.
- Endsley, M. R. (1995). Measurement of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1), 65–84. <https://doi.org/10.1518/001872095779049499>
- Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1), 32–64. <https://doi.org/10.1518/001872095779049543>
- Endsley, M. R. (1996). Automation and situation awareness. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance*. Human factors in transportation (pp. 163–181). CRC Press. <https://doi.org/10.1201/9781315137957-8>.
- European Commission. (2021). *Industry 5.0—Human centric and resilient European industry*. European Commission. Retrieved December 17, 2021, from <https://op.europa.eu/en/publication-detail/-/publication/aed3280d70fe-11eb-9ac9-01aa75ed71a1/language-en/format-PDF/source-search>
- Faccio, M., Granata, I., Menini, A., Milanese, M., Rossato, C., Bottin, M., Minto, R., Pluchino, P., Gamberini, L., Boschetti, G., et al. (2022). Human factors in cobot era: A review of modern production systems features. *Journal of Intelligent Manufacturing*, 1, 1–22.
- Glaessgen, E., & Stargel, D. (2012). The digital twin paradigm for future NASA and U.S. Air Force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference*. <https://doi.org/10.2514/6.2012-1818>.
- Hashimoto, S. (2006). Kansei robotics to open a new epoch of human-machine relationship-machine with a heart. In *ROMAN 2006—The 15th IEEE international symposium on robot and human interactive communication* (p. 1). IEEE.
- Irvine, E. (2013). Measures of consciousness. *Philosophy Compass*, 8(3), 285–297. <https://doi.org/10.1111/phc3.12016>
- Jazdi, N., Ashtari Talkhestani, B., Maschler, B., & Weyrich, M. (2021). Realization of AI-enhanced industrial automation systems using intelligent digital twins. *Procedia CIRP*, 97, 396–400. <https://doi.org/10.1016/j.procir.2020.05.257>
- Kulkarni, S., & Rodd, S. F. (2020). Context aware recommendation systems: A review of the state of the art techniques. *Computer Science Review*, 37, 100255. <https://doi.org/10.1016/j.cosrev.2020.100255>
- Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, 131, 103498.
- Löcklin, A., Jung, T., Jazdi, N., Ruppert, T., & Weyrich, M. (2021). Architecture of a human-digital twin as common interface for Operator 4.0 applications. *Procedia CIRP*, 104, 458–463.
- Löcklin, A., Müller, M., Jung, T., Jazdi, N., White, D., & Weyrich, M. (2020). Digital twin for verification and validation of industrial automation systems—A survey. In *2020 25th IEEE international conference on emerging technologies and factory automation (ETFA)*, 2020 (pp. 851–858).
- Mouret, J. B., & Chatzilygeroudis, K. (2017). 20 Years of reality gap: A few thoughts about simulators in evolutionary robotics. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 1121–1124). Association for Computing Machinery. <https://doi.org/10.1145/3067695.3082052>.
- Müller, M., Jazdi, N., Löcklin, A., Hettich, L., & Weyrich, M. (2022). Adaptive models for safe maintenance planning of CPS. *Procedia CIRP*, 112, 214–219.
- Müller, M. S., Jazdi, N., & Weyrich, M. (2022). Self-improving models for the intelligent digital twin: Towards closing the reality-to-simulation gap. *IFAC-PapersOnLine*, 55(2), 126–131. <https://doi.org/10.1016/j.ifacol.2022.04.181>
- Pairet, È., Ardón, P., Liu, X., Lopes, J., Hastie, H., & Lohan, K. S. (2019). A digital twin for human-robot interaction. In *2019 14th ACM/IEEE international conference on human-robot interaction (HRI)*, 2019 (pp. 372–372). IEEE.
- Reich, J., & Trapp, M. (2020). SINADRA: Towards a framework for assurable situation-aware dynamic risk assessment of autonomous vehicles. In *2020 16th European dependable computing conference (EDCC)*, 2020 (pp. 47–50). IEEE.
- Rizzi, C., Johnson, C. G., Fabris, F., & Vargas, P. A. (2017). A situation-aware fear learning (SAFEL) model for robots. *Neurocomputing*, 221, 32–47. <https://doi.org/10.1016/j.neucom.2016.09.035>

- Romero, D., & Stahre, J. (2021). Towards the resilient Operator 5.0: The future of work in smart resilient manufacturing systems. *Procedia CIRP*, 104, 1089–1094.
- Romero, D., Stahre, J., Wuest, J., Noran, O., Bernus, P., Fast-Berglund, Å., & Gorecky, D. (2016). Towards an Operator 4.0 typology: A human-centric perspective on the fourth industrial revolution technologies. In *Proceedings of the international conference on computers and industrial engineering (CIE46)*
- Ruppert, T., & Abonyi, J. (2020). Integration of real-time locating systems into digital twins. *Journal of Industrial Information Integration*, 20, 100174.
- Ruppert, T., Jaskó, S., Holczinger, T., & Abonyi, J. (2018). Enabling technologies for Operator 4.0: A survey. *Applied Sciences*, 8(9), 1650.
- Sahlab, N., Kamm, S., Müller, T., Jazdi, N., & Weyrich, M. (2021). Knowledge graphs as enhancers of intelligent digital twins. In *2021 4th IEEE international conference on industrial cyber-physical systems (ICPS)*, 2021 (pp. 19–24). IEEE.
- Salerno, J. (2008). *Measuring situation assessment performance through the activities of interest score*. IEEE.
- Salfinger, A. (2020). Reinforcement learning meets cognitive situation management: A review of recent learning approaches from the cognitive situation management perspective. In *2020 IEEE conference on cognitive and computational aspects of situation management (CogSIMA)*, 2020. IEEE.
- Sirithunge, C., Ravindu, H. M., Bandara, T., Buddhika, A. G., Jayasekara, P., & Chandima, D. P. (2019). Situation awareness for proactive robots in HRI. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2019. IEEE.
- Solms, M. (1997). What is consciousness? *Journal of the American Psychoanalytic Association*, 45(3), 681–703; discussion 704–778. <https://doi.org/10.1177/00030651970450031201>.
- Thorvald, P., Fast Berglund, Å., & Romero, D. (2021). The cognitive Operator 4.0. In *International conference on manufacturing research*, 2021 (Vol. 15, pp. 3–8). IOS Press.
- Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2017. IEEE.
- West, T. D., & Blackburn, M. (2018). Demonstrated benefits of a nascent digital twin. *INSIGHT*, 21(1), 43–47. <https://doi.org/10.1002/inst.12189>
- White, F. E. (1991). *Data fusion lexicon* (Vol. 1(1), pp. 1–16). Joint Directors of Labs.
- Wickens, C. D. (2008). Situation awareness: Review of Mica Endsley's 1995 articles on situation awareness theory and measurement. *Human Factors*, 50(3), 397–403. <https://doi.org/10.1518/001872008X288420>
- Xie, X., Sun, S., Chen, H., & Qian, J. (2018). Domain adaptation with twin support vector machines. *Neural Processing Letters*, 48(2), 1213–1226.
- Yujian, L., & Bo, L. (2007). A normalized Levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 1091–1095. <https://doi.org/10.1109/tpami.2007.1078>
- Yusuf, S. M., & Baber, C. (2022). Formalizing distributed situation awareness in multi-agent networks. *IEEE Transactions on Human-Machine Systems*, 1, 1–10. <https://doi.org/10.1109/thms.2022.3142109>
- Zhang, X., Chan, F. T., Yan, C., & Bose, I. (2022). Towards risk-aware artificial intelligence and machine learning systems: An overview. *Decision Support Systems*, 159, 113800. <https://doi.org/10.1016/j.dss.2022.113800>
- Zhao, W., Queralta, J. P., Qingqing, L., & Westerlund, T. (2020). Towards closing the sim-to-real gap in collaborative multi-robot deep reinforcement learning. In *2020 5th International conference on robotics and automation engineering (ICRAE)*, 2020 (pp. 7–12). IEEE.
- Zilly, F. (2023, January 26). Data management—For the development of highly automated driving functions.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.