Toward Adaptive Robot Behavior for Interdependent Human-Robot Teams

Pietras, Emilia; Hall, Stephanie ; Kiefer, Bernd; Langaa, Jeppe; Borzone, Guglielmo; Dhanda, Mandeep; Kramberger, Aljaz; Dhokia, Vimal; Krüger, Norbert; Bodenhagen, Leon

Publication date:
2024

Document version:
Other version

Citation for pulished version (APA):
Pietras, E., Hall, S., Kiefer, B., Langaa, J., Borzone, G., Dhanda, M., Kramberger, A., Dhokia, V., Krüger, N., & Bodenhagen, L. (2024). *Toward Adaptive Robot Behavior for Interdependent Human-Robot Teams*. Paper presented at ICRA 2024 workshop: Dynamic Duos: Human-Robot Co-Worker Adaptation in Manufacturing. , Yokohama, Japan.

Toward Adaptive Robot Behavior for Interdependent Human-Robot Teams

Emilia Pietras*, Stephanie Hall, Bernd Kiefer, Jeppe Langaa, Guglielmo Borzone, Mandeep Dhanda
Aljaz Kramberger, Vimal Dhokia, Norbert Krüger, Leon Bodenhagen

I. Introduction

In manufacturing today, a gap persists in the implementation of safe and efficient human-robot teams. To implement robot systems that truly collaborate with human teammates, such systems must facilitate adaptive high-level robot action planning. This adaptive robot behavior should both account for the goal of the collaborative task and the characteristics of their human teammate. Adapting to the human would not only take into account interpersonal differences but also the changes in the human's physical and mental state throughout the collaboration.

In this paper, we will first present the related work in Sec. II, followed by our proposed model in Sec. III, which describes the modules needed to implement adaptive human-robot interaction (HRI) systems. The model consists of three main elements: 1) the human-robot team, 2) the knowledge base, and 3) the high-level action planning. These main parts each contain two or more subelements, which will be described in further detail. In Sec. IV, the method is put in the context of three different use cases: learning from virtual demonstrations, assembling and fixing components, and object scanning.

II. Related Work

A. The Use of Ontologies in Robot Systems

The integration of robots in an environment where they will be working closely with humans brings up issues such as scheduling, safety, and trust between the human-robot team due to the stochastic nature of humans and the complexity of the interaction. Efficient collaboration between the human and robot requires both the human and robot to understand what each other is doing and have an awareness of what else may be going on in the environment. Therefore, a way to structure and store this large amount of often abstract information and identify how it is interlinked is needed.

Ontologies can help fuse and contextualize the large amount of information obtained from sensing devices, algorithms, etc. during the process. General ontologies for robotics have been developed, such as the core ontology for robotics and automation (CORA) [1], with additional research extending this to focus on definitions for positioning, orientation, and robot poses [2].

Numerous ontologies have been developed to tackle autonomous robotics and HRI. Lemaignan et al. [3] focus on the cognitive skills needed in the OpenRobots ontology (ORO), considering abilities such as symbolic reasoning, theory of mind, working memory, desires and experiences as important considerations. Robot Task Planning Ontology (RTPO) focuses on separate task, environment, and robot ontologies [4].

However, some of these approaches focus mainly on the robot behavior, limiting the human to simple communication and ignoring the possibility of unpredictable events or plan changes due to human behavior [5]. Many of these ontological approaches to model the concepts involved in manufacturing, robotics, etc. related applications tend to have a low-level inclusion of the collaboration between the human and robot. David et al. [6] aim to address these limitations in their ontology, collaborative agents for manufacturing ontology (CAMO). They consider different modalities for communication, such as vision, whereas speech is commonly used in previous human-robot collaboration (HRC) applications. A far greater focus is on social attributes and teamwork and consideration is given to the mental attitudes of the human workers in the interaction.

B. Knowledge Processing Systems for Robotics

For flexible and general implementation of robot behavior, robots must be equipped with the ability to reason and infer action parametrizations. A well-known knowledge processing system made to achieve this is KnowRob [7], a system made for autonomous personal robots. It comprises encyclopedic knowledge, an environment model, action-based reasoning, and human observations. Furthermore, it allows for querying continuous sensor data and uses a knowledge base to which observations can be loaded into. An extension of this system, KnowRob 2.0, has increased capabilities including an inner-world knowledge base [8]. This knowledge base consists of a simulated reconstruction of the robot's environment with physics simulation and vision capabilities, letting the robot reason about actions and gain intuitive physics knowledge. Furthermore, this knowledge processing system works as a part of an even larger cognitive architecture, CRAM [9], addressing the challenge of robots performing general manipulation tasks. The knowledge base in our proposed system is in accordance with existing systems as it has been developed with an ontology reuse approach. Meanwhile, more specifically, our proposed system contributes to exisitng works with its emphasis on higher-level task planning accounting for a human teammate.

C. HRI and Cobots in Manufacturing

HRI can be categorized into human-robot coexistence, cooperation, and collaboration [10]. HRC commonly encompasses the human and the robot sharing their workspace and task. It is thus more involved and proximate than cooperation and coexistence, as they do not require a shared workspace or manipulating the same objects simultaneously [10]. The need for HRC in manufacturing is particularly evident in tasks where automated systems are unable to execute complex processes without humans and is also a prospect of improving efficiency and reducing human workload [10].

However, a gap remains in implementing HRC in manufacturing today [11]. One of the main challenges within HRC in manufacturing is safety, which comprises physical safety (e.g., avoiding collisions) and psychological safety, focusing on minimizing potential discomfort caused by the robot [12]. Human-related adaptation (e.g., basing the robot's strategies on the human operators' skill and experience level) may assist in keeping the human in a desired physical and psychological state. Nevertheless, this remains difficult to develop and implement due to interpersonal differences and human unpredictability [11]. Our proposed method aims to contribute to the work within human-related adaptation through the knowledge base and high-level robot action planning.

III. Method

In the following, we describe our system in which a robot can adapt to the user's preferences and internal state during collaboration. The full system is illustrated via a block diagram, shown in Fig. 1. In sum, as the human-robot team performs a collaborative task, some form of interaction occurs (communication, manipulation of shared objects, physical interaction, etc.). Meanwhile, the internal state of the human user is inferred from physiological and behavioral data. The RDF store[1], which inherits domain knowledge from the ontology, is responsible for storing this data and updating the user profile. Based on the collaborative task and the inferred state, a decision framework is used to select the appropriate strategy. Lastly, a Behavior Tree controls the flow of the robot's task while facilitating reactive task switching.

A. The Human-Robot Team: User and Robotic System

1) User: the user interacts with the robotic system through several modalities:

- The robot arm: e.g., physical interaction, manipulation of shared objects.
- The *multimodal interface*: e.g., Graphical User Interface (GUI), vision- or audio-based communication.
- Indirect interaction: the human and the robot coexisting in the same workspace implies the presence of indirect interaction. One example is proxemics, which necessitates that the robot plans for collision avoidance and respecting the human space.

[1]HFC: https://github.com/bkiefer/hfc-thrift

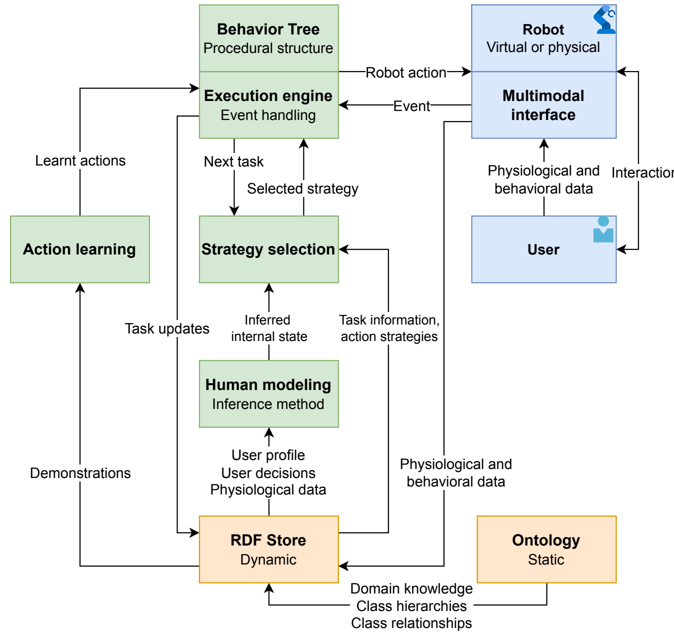

Fig. 1. Block diagram presenting a model for a robot system that adapts to the user's needs, preferences, and internal state during collaboration. During the *interaction* between the robot and the user, the *RDF store*, which incorporates class information from the ontology, is updated based on user data, and the *human modeling* block infers the user's internal state. Based on the task and the user, the optimal robot strategy is selected and executed.

In addition to the physical and communicative aspects of the collaboration, the *multimodal interface* may encompass physical, visual, or audio-based sensors to record physiological and behavioral data. These recordings serve as observations to model the human's internal state. It is often also relevant to account for recurring users, meaning that long-term information about their preferences and past sessions with the system must be kept in the *RDF store*. Here, a database is dynamically updated to store information about the user and the session.

2) Robot: in our proposed framework, the robotic system is implemented as a composite of the robot proper (a simulated or real robot) and a *multimodal interface*. In either case, the *Behavior Tree* sends the next high-level *robot action* to be executed by the robot (e.g., moving to a specified location or sharing information with the human teammate).

3) Multimodal interface: this part of the robotic system also interacts with the user as it is the module that handles communication. For instance, it may be a GUI from which the user can make decisions and the robot can share information. The interface may involve various forms of communication, e.g., speech, gestures, and expressions. Apart from communication, the *multimodal interface* sends *physiological and behavioral information* about the user to the *RDF store*, from the real-time output of various monitoring devices. Since it is part of the robotic system, the *multimodal interface* receives *robot actions* from the *Behavior Tree*, such as speech output. Actions may result in some perception output, e.g., a scanning action that results in a point cloud. Such outputs and other events, such as *user decisions*, are sent to the *execution engine*, which handles all outputs from the user and the robot.

B. The Knowledge Base: Ontology and RDF Store

1) Ontology: The knowledge base in this framework uses information gained from other blocks of the system to infer contextualized knowledge about the current state of the process. This can be used to determine the current cognitive and physical state of the worker or robot, whether certain tasks can be actionable, which events may be occurring, etc.

The role of the ontology in this system is to store a formal, structured map of all possible domain knowledge, including the key concepts, properties, relationships, and axioms. This model consists of a formal way to describe both the physical aspects of the human and robot, their skills and capabilities, and additionally for the human, their previous experience, behavior, social and emotional features. Use case objects, task specifications, and actions are also contained in the ontology, linking and constraining these concepts to the more abstract features of the human and robot.

2) RDF Store: The *RDF store* is a dynamic knowledge base and customizable reasoner containing information about all interactions, such as a given user's previous preferences. The *RDF store* makes use of world knowledge from the ontology, information about the task progression from the *execution engine*, and real-time physical and behavioral information about the human user from the *multimodal interface*. All task and user information is updated in the RDF store database and the relevant updates are passed on to the *human modeling* block. Since the store not only contains the present state, but also previous states by assigning each triple the *transaction time* when it was added to the database. In this way, a long-term history is kept that opens up lots of possibilities for personalized and social behaviour through adaptation.

C. The High-Level Action Planning

1) Human modeling: the *human modeling* block infers the internal state of the user based on data collected from the *user profile* (if it is a recurring user) and the data collected on-the-fly during the task. To model the state of the human, we may want to know something about their *user profile* (e.g., what is their skill level? What are their previous preferences?), *user decisions*, and *physiological data*. The human model receives these observations about the user from the *RDF store* throughout the duration of the collaborative task. Based on these observations, the output of the *human modeling* block is the inferred state or preference in the form of a probability. The probability distribution is sent to the *strategy selection* block.

2) Strategy selection: this block takes input from the *Behavior Tree*, informing the *next task* which is the upcoming action the robot should execute. The block also receives input from the *human modeling* block, providing the *inferred internal state* of the user, and the *RDF store*, providing possible action strategies and relevant task information. The purpose of the *strategy selection* block is to select the appropriate robot action strategy given the task and the user's state. The strategy may meet a condition that is always checked by the *Behavior Tree* or it can change the outcome of the next *robot action*, varying the execution speed, level of proximity, etc. A practical example is the following: from the *human modeling* block the inferred state of the user is a state of high fatigue. The highest-reward option found by the *strategy selection* method is to suggest a break, so the condition "SuggestBreak?" is updated to *True* and is read by the *execution engine*.

3) Action learning: this module is responsible for using the trajectories demonstrated by the user for robot learning. These trajectories could come from the human moving the physical robot in free-drive mode or potentially from moving the robot in Virtual Reality (VR) via controllers, as further described in Sec. IV-A. These trajectories would then be kept to the *RDF Store* and provided to the *action learning* module.

4) Execution engine: the purpose of the *execution engine* is to serve as an abstraction from the *Behavior Tree*, making sure the *Behavior Tree* does not itself take care of directly receiving and managing input from the human-robot team or the *strategy selection*. The *Behavior Tree* should be as simple as possible as it is only a procedural structure and simultaneously managing events will make it more complex. Thus, the *execution engine* handles the output from the *multimodal interface* about any interaction or task events. Such events may include input from the user or the status of ongoing *robot actions*. A practical example is the following: the human teammate abruptly orders the robot to stop in the middle of a task. This event is handled on-the-fly in the *execution engine* and passed on to the *Behavior Tree*, which will immediately request the robot to halt.

5) Behavior Tree: this block represents the state machine that allows the robot to switch between tasks and trigger *robot actions* based on the hierarchical, sequential, and conditional structure of the tree. The *Behavior Tree* takes information about the *selected strategy* and the task events handled by the *execution engine* to then determine and trigger the next *robot action* to be performed by the robot or communicated through the *multimodal interface*.

IV. Examples of Use Cases

A. Use Case 1: Learning from virtual demonstrations

In the industrial world, it is usually possible to access 3D models of the parts used within a certain task. Furthermore, the workcells are often meticulously set up and may even be calibrated. This makes it easy to replicate within a virtual scene and to simulate the task. Recreating the industrial task in a virtual setting allows the use of VR to traverse the scene and inspect the task or to interact with the task and the robot in a variety of ways. One way to interact with the robot is to use the controllers of the VR setup within the simulation to demonstrate a task by letting the tool center point (TCP) of the robot follow the target of the controllers. This allows the user to demonstrate a trajectory without being in the vicinity of the robot, which lowers the risk level and lets the human operator demonstrate tasks that are normally too heavy for the human body. The usual method of kinesthetic teaching does not allow the human operator to stay in a safe area of the workspace and is usually not appropriate for tasks requiring heavy loads. Furthermore, it is possible to demonstrate a robot task within the correct action and state space as there is no external disturbances in the form of a human operator. This eases the learning as the training domain is the same as the test domain. Combining an off-policy Reinforcement Learning (RL) algorithm that includes an expert replay buffer for the demonstrated data, along with the demonstrations from a VR headset, one can demonstrate an arbitrary task while allowing the operator to correct future behavior by recording new demonstrations. The robot behavior will be bootstrapped from the initial demonstrations of the operator and will develop as the robot gets experience by carrying out the task. The continuous learning of the robot can be corrected by the human operator by recording new demonstrations, as the expert demonstrations are more substantial to the robot than its own experience.

B. Use Case 2: Assembling and fixing components

One of the use cases for this HRI system is the assembly of a mechanical component which will be assembled into a ring, used for the construction of jet engines (see Fig. 2). In this task, two metal parts have to be precisely aligned and joined together. The robot and the human will cooperate in a shared workspace, with the human assessing the quality of the assembly and adjusting it to achieve the best result. A simulation of the workspace will be available to the operator, making it possible to review the robot's next course of action before issuing the command to the real system. All of the actions performed by the robot are activated by the user through a graphical user interface (GUI).

The task can be broken down as follows:

1) The robot picks up the first component and moves it above one of the two fixtures, as chosen by the user.
2) The robot attempts to insert the component into the fixture unassisted.
3) In case of failure, the robot asks the human to help place the component and the robot enters free-drive mode. The human helps the robot place the component and notifies it when the part is in place.
4) When prompted by the user, the robot picks up the second component and holds it above the second fixture.
5) The robot repeats steps 2, 3 and 4.
6) The robot picks up the joint plaque and holds it in place, attempting to align the holes of the plaque with the holes in the left and right parts.
7) The robot enters free-drive mode, allowing the user to make final adjustments to further align the parts.
8) The human holds the second plaque and completes the task by riveting the parts.

In the current setup, the human communicates with the robot through a GUI, which could be extended with a voice interface, increasing the speed and ease of communication. Furthermore, to integrate the robot system with adaptive behavior, a learning algorithm could be implemented in the part where the robot

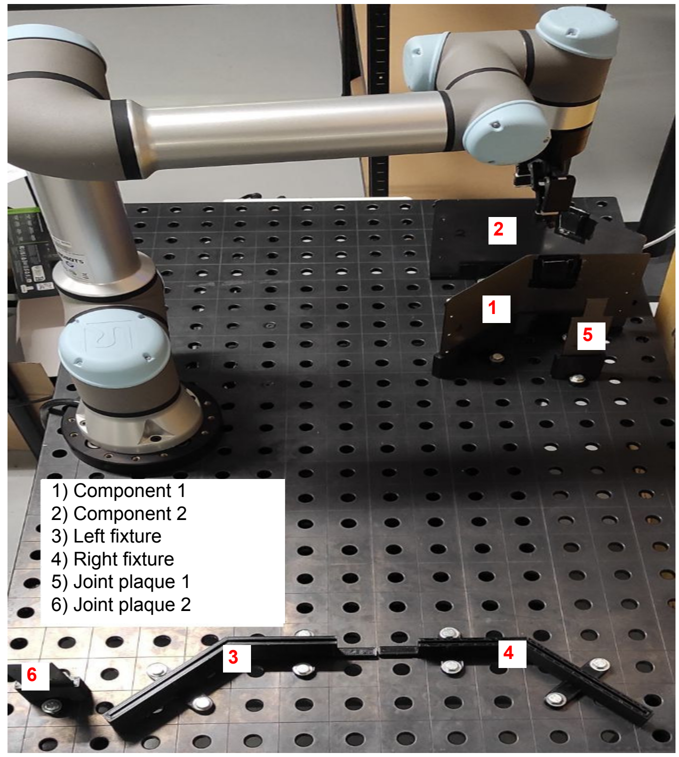

Fig. 2. Image of the physical setup with a UR robotic arm and the components used for the assembly of the ring. The two components are placed into the left and right fixtures and the two sides are joined using the joint plaques. Once the human-robot team has finished the assembly, the human rivets the parts to fix the components in place.

tries to place the object by itself: the human guides the robot, which has entered free-drive mode while the robot registers and learns from the trajectory that the human made it follow. Additionally, more control could be provided to the human. In such a proxmiate HRI task, the human worker could feel intimidated by the vicinity of the robot. A way to make the robot more trustworthy would be to provide the human with controls such as adjusting the robot's proximity so that an inexperienced user could take their time to familiarize themselves with the system.

C. Use Case 3: Object Scanning

Another use case for the proposed adaptive HRI system is an object scanning case, in which the robot scans an object with a scanner in hand while supervised by a human. Our setup is in virtual reality (VR), meaning that the human is presented with a simulated worktable along with the robot and the object. A GUI is also present in the VR scene, providing the human with guidance as well as an interface for making decisions. To interact with the GUI, the human uses the available VR controllers. An example of the simulated setup is shown in Fig. 3. Independent of the user's profile or state, the general interaction is the following:

1) The robot determines the object dimensions (height, width, and length).

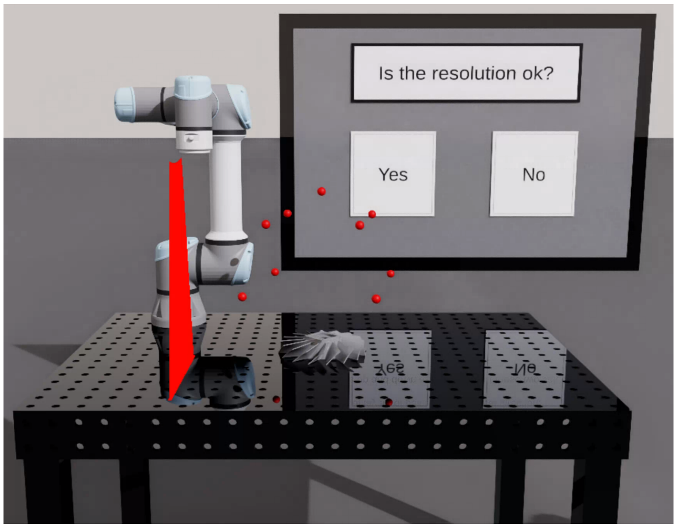

Fig. 3. Example of setup with simulated robotic arm and a small object on a worktable. The red beam from the robot's end-effector shows where the robot is scanning while the red dots in space are the scanning point to which the robot will move to scan the object. In the background is a display showing a simple GUI.

2) The robot suggests an initial scan plan (a set of poses for scanning).
3) The human accepts the scan plan or makes modifications (increasing/decreasing the number of poses or adding poses manually).
4) The robot scans the object and displays the final scan to the user.
5) The human decides if the scan is complete (done), incomplete (goes back to step 3 to modify the scan plan), or failed (goes back to step 1 to start over the scanning process).

However, beyond the sequence just described, several human variables should result in a different experience for the user. One example is that the robot should adapt its instructions to the user's level of experience with the task. The system should introduce inexperienced users to the workspace, the terminology used, the goal of the task, and the user's options for quality check (determining scan complete, incomplete, or failed). In contrast, an experienced user may become bored or frustrated from receiving unnecessary instructions. To reach adaptive communication, the system benefits from its RDF store which stores the user's session history. Storage of previous user sessions will allow a database of user profiles, informing the system about the user's previous preferences (e.g., choosing to skip instructions in past sessions) and level of complacency (e.g., rate of accepting system suggestions for the scan plan or accepting the final scan). This could also allow the system to suggest scan plans according to the user's previous preferences.

V. Discussion

A. Contributions

Our contributions include the presentation of a model for general robot systems in various mediums, specifically with a focus on true collaborative tasks that require mutual adaptation in a human-robot team. The model's use of an ontology allows for structuring the large amount of often abstract information about the interaction and the task environment and identifying how this information is interlinked. Meanwhile, the dynamic knowledge base (the *RDF store*) is separated but inherits from the ontology, storing information from sensors and observed behaviors. In addition to the collaborative aspect of robot action planning, this storing of information also accounts for long-term interaction. Moreover, the proposed model comprises modules responsible for robot action planning and execution in a way that allows for variation in complexity (i.e., variation in the complexity of the inference and strategy selection methods). Finally, the described use cases lay the groundwork for the future implementation of adaptive robot teammates in real and virtual mediums.

B. Challenges

Although the proposed system contains the modules needed for an adaptive robot system, it also allows for a great variation. For the practical implementation, one will still need a clear understanding of the desirable human states and the appropriate robot strategies to accommodate for interpersonal differences and changes in the human's state. On the same note, there needs to be an overall accounting for human safety and unpredictability. Thus, to ensure that the robot adapts its behavior appropriately to overall benefit the collaboration, suitable methods for human state inference and robot action planning must be determined and applied.

VI. Conclusion and Future Work

In this paper, we have proposed a system facilitating the implementation of human-robot teams with mutual adaptation. It allows for the robot to account for and adapt its behavior to the human teammate while the latter familiarizes themselves with the task and gains experience. The proposed model can apply to real robot setups as well as VR setups, which in particular show potential for smooth multimodal interaction in a minimal setup. Furthermore, it aims to improve generalizability to other HRI tasks and extended knowledge about the human worker through the ontology-based framework. In future work we plan to implement a robotic system in VR that adapts its action strategies based on a given user's interaction history, exemplifying the architecture framework described in this workshop paper.

Acknowledgment

This work has been funded by the EU project Fluently (Grant agreement ID: 101058680) and supported by the Industry 4.0 lab at the University of Southern Denmark.

References

[1] E. Prestes, J. L. Carbonera, S. R. Fiorini, *et al.*, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, no. 11, 2013.

[2] J. L. Carbonera, S. R. Fiorini, E. Prestes, *et al.*, "Defining positioning in a core ontology for robotics," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013.

[3] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human—robot interaction: An implementation," *Artificial Intelligence*, vol. 247, 2017. DOI: 10.1016/j.artint.2016.07.002.

[4] X. Sun, Y. Zhang, and J. Chen, "RTPO: A domain knowledge base for robot task planning," *Electronics*, vol. 8, no. 10, 2019. DOI: 10.3390/electronics8101105.

[5] J. I. Olszewska, M. Barreto, J. Bermejo-Alonso, *et al.*, "Ontology for autonomous robotics," in *2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN)*, IEEE, 2017.

[6] J. David, E. Coatanéa, and A. Lobov, "Deploying OWL ontologies for semantic mediation of mixed-reality interactions for human–robot collaborative assembly," *Journal of Manufacturing Systems*, vol. 70, 2023. DOI: 10.1016/j.jmsy.2023.07.013.

[7] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoğlu, and G. Bartels, "Know rob 2.0 — a 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018. DOI: 10.1109/ICRA.2018.8460964.

[8] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoğlu, and G. Bartels, "Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018.

[9] M. Beetz, G. Kazhoyan, and D. Vernon, "The cram cognitive architecture for robot manipulation in everyday activities," *arXiv preprint arXiv:2304.14119*, 2023.

[10] S. Hjorth and D. Chrysostomou, "Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly," *Robotics and Computer-Integrated Manufacturing*, vol. 73, 2022.

[11] R. R. Galin and R. V. Meshcheryakov, "Human-robot interaction efficiency and human-robot collaboration," in *Robotics: Industry 4.0 issues & new intelligent control paradigms*, Springer, 2020.

[12] W. Xian, K. Yu, F. Han, L. Fang, D. He, and Q.-L. Han, "Advanced manufacturing in industry 5.0: A survey of key enabling technologies and future trends," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, 2023.