2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Ontology Based AI Planning and Scheduling for Robotic Assembly

Jingyun Zhao[1], Birgit Vogel-Heuser *IEEE Fellow*[1], Jicong Ao[2], Yansong Wu[2], Liding Zhang[3], Fandi Hartl[1], Dominik Hujo[1], Zhenshan Bing[3], Fan Wu[2], Alois Knoll[3], Sami Haddadin[2], Bernd Vojanec[4], Timo Markert[5], André Kraft[6]

Abstract— **The rising demand for customized products necessitates the integration of multiple robotic systems, underscoring the need for advanced production planning and scheduling. This paper introduces an ontology-based, artificial intelligence-enhanced method for dynamic task planning and scheduling, aimed at improving the efficiency of production process, reducing machine downtime, and consequently increasing throughput in assembly operations. Designed to generate and execute feasible production plans dynamically, this method minimizes manual planning and scheduling efforts. We evaluate its effectiveness using two gear assembly use cases with various robot skills, highlighting its flexibility in planning and scheduling and its contributions to the evolution of smart manufacturing. The method's adaptability suggests its applicability across diverse smart factory environments.**

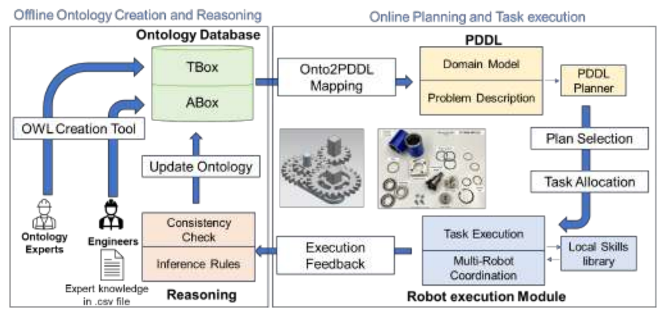

Fig. 1. Concept overview: automatic generation of PDDL from ontologies, execution of selected plan on robot systems and update execution results in ontology

I. INTRODUCTION

Planning and scheduling of production process in manufacturing have long been challenging tasks, particularly in the context of Industry 4.0 where the demand for individualized products is increasing. This demand necessitates quick responses and efficient adjustment in production process or setups to cater to individualized product specifications. In assembly environments, often characterized by the collaboration of multi-robot systems, each assigned a specific process step for individual products, the efficient allocation and scheduling of resources and processes become imperative.

Systematic process planning and scheduling can substantially reduce machine downtime and enhance throughput. However, traditional planning and scheduling methods are not efficient to address the dynamic complexities inherent in assembly operations. In this paper, we introduce an ontology-based approach integrated with an Artificial Intelligence (AI) planning tool, the Planning Domain Definition Language (PDDL), for automatically generating and executing assembly plans. The ontology serves as a comprehensive knowledge base, modeling domain-specific knowledge within the production sector and detailing specific assembly properties. Based on the knowledge base and current states of robots, we generate production plans utilizing PDDL. To evaluate our method, we applied it to two gear assembly use cases within a laboratory setting: one involving multi-robot systems in a traditional large-scale production assembly line, and another involving a single robot equipped with various grippers in flexible assembly setups for small-scale production. The results highlight the efficiency and generalizability of our method. Future research will aim to expand this method for real-time planning in additional use cases.

The contribution of this paper lies in the development of a closed-loop method for automatic and dynamic assembly process planning, applicable to both traditional assembly line operations and flexible assembly station setups, as shown in Fig.1. The key points include:

- Extending the intermediate ontology of products, processes, and resources (PPR) [1] by adding extra relationships and applying the ontology to depict the interactions between grippers and objects.
- Facilitating the automatic generation of plans without manual intervention, significantly reducing manual effort.
- Integrating the feedback from the actual execution of these plans into the ontology, enabling continuous knowledge updating and enhancement.

II. Background and Related works

Ontology, defined as a "formal, explicit specification of a shared conceptualization" [2], organizes domain knowledge

[1]Chair of Automation and Information Systems, Technical University of Munich, 85748, Garching, Germany {jingyun.zhao, vogel-heuser, fandi.bi, dominik.hujo}@tum.de
[2]Chair of Robotics and Systems Intelligence, Technical University of Munich, 80992, Munich, Germany {jicong.ao, yansong.wu, f.wu, haddadin}@tum.de
[3]Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich, 85748, Garching, Germany liding.zhang@tum.de, knoll@in.tum.de
[4]WITTENSTEIN SE, Corporate Research & Development, 97999, Igersheim, Germany bernd.vojanec@wittenstein.de
[5]Resense GmbH, 63911, Klingenberg, Germany timo.markert@resense.io
[6]BMW AG, Digitalisierung Shopfloor, 80939, Munich, Germany andre.kraft@bmw.de

using a structured vocabulary and delineates relationships among terms. It consists of a TBox (terminology component), detailing concepts and relationships, and an ABox (assertion component), presenting specific instances associated with the conceptual models outlined in the TBox [3]. In manufacturing, ontologies are increasingly adopted for their effective domain knowledge representation, enabling abstract-level flexibility and reusability [4], and facilitating new knowledge discovery through automated reasoning and inference [5]. Numerous well-established ontologies are available in the manufacturing and robotics domain, providing options for assembly use cases. Upper ontologies like DOLCE [6] and SUMO [7] offer broad domain applicability. MASON delineates the manufacturing-specific PPR framework [8], while IEO [1] bridges DOLCE with domain-specific ontologies. CaSkMan [9] and CORA [10] align various ontologies with industry norms, improving robotics and automation compatibility. Additionally, specialized assembly ontologies, such as the kitting ontology [11], [12], MaRCO [13], and the Assembly Process Ontology [14], [15], delve into assembly-specific tasks, enhancing task execution and lifecycle product structure insights.

The PDDL, designed to standardize AI planning language, aims to represent and resolve classical planning and scheduling problems [16]. PDDL has evolved through five versions, introducing new features in each. PDDL 1.2 introduced clear domain-problem separation based on predicate logic [16]. Crucially, PDDL 2.1 extended PDDL 1.2 numeric fluents and durative actions [17]. Although PDDL has seldom been applied to real-world production problems, it can significantly support the manufacturing process planning e.g., enabling discrete problem modeling and plan generation [18], and can be aligned with IEC 62264 standards to improve assembly planning [19]. Although beneficial for standardized planning and enhancing cross-study usability, PDDL limits expressiveness in addressing complex problems, especially in the case of temporal planning [17], [19]. The diversity of planners also presents interoperability challenges, impacting flexibility across planning environments. Thus, using PDDL requires tailored adaptation for varied planning contexts.

Planning and scheduling synergies to organize tasks; the former outlines task sequences while the latter assigns resources and timing for execution [4]. Ontology and PDDL both facilitate domain knowledge formalization, beneficial for effective task allocation and scheduling. Ontology contributes semantic richness to PDDL models, refining the clarity and depth of planning problem representations. In practice, automated planners leverage ontological insights for improved decision-making and domain analysis, resulting in optimized search and inference mechanisms. Despite their longstanding development, the integration of ontology and PDDL in real-world production remains rare. A detailed review is provided in Table I, illustrating the application and utility of ontology and PDDL across various settings and challenges. We can conclude that the combined implementation of ontology and PDDL for automated planning and scheduling in dynamic environments, especially in robotic and assembly scenarios with complex PPR relations, is underexplored. Assembly planning often concentrates on the roles of robots and processes, neglecting essential components such as grippers and objects, and lacking in-depth scheduling and execution time analysis. Furthermore, the integration of ontology to its fullest potential and the effectiveness of PDDL in handling complex scenarios have not been thoroughly investigated and utilized.

III. Ontology Based Planning Method

This section outlines our method, encompassing four components: ontology construction, ontology to PDDL transformation, execution of the planning results, and ontology-based reasoning.

A. Concept Overview

Our method of automatic, dynamic task planning and scheduling employs a domain-specific ontology, initially designed by ontology experts (for TBox) and enriched by engineers (for ABox), as depicted in Figure 1. This foundational ontology facilitates the extraction and automatic transformation of necessary application knowledge into PDDL, guided by specific rules tailored to different planning and scheduling scenarios. Optimal planning outcomes are identified through a cost function that evaluates results from different PDDL planners, which are equipped with numeric and durative action capabilities. The selected plan is then applied within robotic systems, which involve coordination among multiple robots or the use of multifunctional grippers. These plans are on an abstract level, with detailed task execution deferred to the robots' internal planning systems. Post-execution and plan outcomes are reincorporated into the ontology, enhancing the system's capacity for inference, consistency validation, and the discovery of new insights, thereby ensuring ongoing improvement of the domain ontology. While ontology construction and reasoning are conducted offline due to their time-intensive nature, the planning component operates online to allow near real-time responses.

B. Ontology Construction

In our method, we select an intermediate ontology instead of upper or application-specific variants to circumvent the constraints presented by PDDL. Employing this intermediate ontology alongside PDDL for high-level, abstract task planning in manufacturing enables us to evade the extensive generalization of planning scenarios and the restricted scope of application-specific ones, ensuring our method is both adequately comprehensive and appropriately focused.

The selected ontology, as shown in Figure 2, is the IEO [1] linked to DOLCE, which features the PPR structure within the manufacturing domain. The IEO characterizes resources, such as robots, by their capabilities to sequentially perform processes, transforming raw materials into finished products, guided by quality indicators or quales. These quales encompass resource usage rate, execution time, and throughput. Furthermore, the ontology is extended with definitions of skills specific to the robotics field [30]. It

TABLE I

AN OVERVIEW OF USING ONTOLOGY AND PDDL FOR TASK PLANNING AND SCHEDULING

Author	Year	Ref.	Stand. Onto.	Onto. Purpose	Auto. PDDL gen.	Dynamic Scheduling	Application
Balakirsky et al.	2013	[11]	CORA	DB	+	+	Kit building
Wickler et al.	2014	[20]	o	DB	+	-	Container shifting
Papadimitriou et al.	2014	[21]	o	DB	-	+	Adaptive mission planning
Kootbally et al.	2015	[12]	CORA	DB	+	+	Kit building
Riccardo et al.	2018	[22]	-	DB, IC	-	-	Robot manipulation with articulated Objects
Silva et al.	2020	[23]	CORA, SUMO	DB, IR	-	-	Bin picking
Ruiz et al.	2022	[24]	o	DB	+	-	Robotic manipulation
Joo et al.	2022	[25]	-	DB, KM	-	+	Robotic navigation
Ruiz-Celada et al.	2023	[26]	-	DB, IR, KU	+	o	Robotic manipulation
Hoebert et al.	2023	[27]	-	DB	+	+	Assembly
Hoebert et al.	2023	[28]	-	DB, DM, CC, IR	+	+	Antenna amplifier disassembly
Hoebert et al.	2023	[29]	CORA	DB, IR	+	+	Robot skill execution and learning
Ours	2024	-	DOLCE	DB, KM, IC, IR, KU	+	+	Gear assembly

"+": supported; "o": partially supported or has the potential to be supported; "-" not supported. DB: database, IC: inconsistency checking, KM: knowledge maintenance, DM: decision making, IR: inference and reasoning, KU: knowledge update.

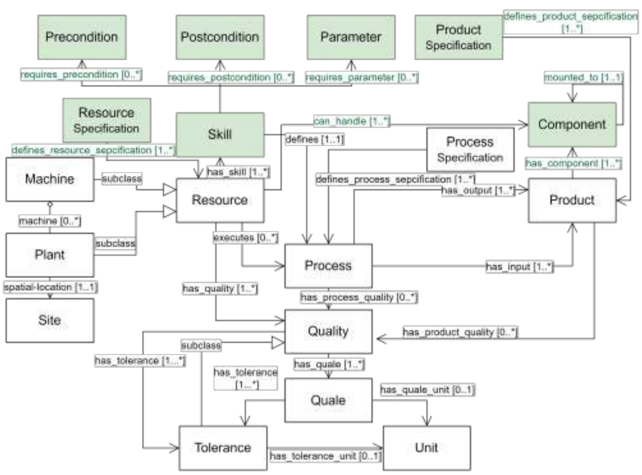

Fig. 2. Extended intermediate ontology. New Classes and instances are marked in green.

also facilitates the inclusion of components within products, incorporates classes specification of resources and products, and delineates the interactions between resources and products. Notably, this paper places particular emphasis on the intricate relationships between grippers (resource) and objects (product).

C. Ontology to PDDL Transformation

In our method, we establish transformation rules among ontologies, PDDL domain and problem files, adapting to different application scenarios. In this research, we employ PDDL version 2.1, integrating numeric fluents and durative actions crucial for task planning that involves execution time characteristics. The mapping scheme is detailed in Table II.

D. Planning Generation and Execution

PDDL planners generate planning outcomes, from which the optimal plan P is chosen based on calculating each plan's cost (C), considering criteria such as execution time (T),

TABLE II

AN OVERVIEW OF ONTOLOGY TO PDDL MAPPING SCHEME

Ontology or robot state	PDDL Domain	PDDL Problem
TBox		
Class - Resource	Types	
Object properties - locates	Predicates	
Data properties - Execution time	Functions	
Axioms - Specific robots can only handle specific objects with skills	Actions	
ABox		
Instances - Robots, Grippers, Products		Objects
Specifications - Final products		Goal state
Robot state		
Current robot state		Initial state

sequence steps (S), and resource usage (R), as delineated by the formula:

$$P = \arg\min_{i} C_i \tag{1}$$

$$C_i = w_t \cdot T_i + w_s \cdot S_i + w_r \cdot R_i, \tag{2}$$

where w_t, w_s, and w_r represent weights assigned to each criterion.

Building on the framework outlined in [30], a task is structured around one or more skills, with each skill typically comprising one or more movement primitives. Owing to this design, the open questions at the task level lie in choosing the proper task. Whereas, primitive-level planning involves too many concrete factors, *eg.*, finding a fitted control architecture, picking suitable movement primitives, and choosing the proper parameters for each primitive. Considering all these overly specific factors always results in an unnecessary excessive burden on the computation power. Therefore, in this work, the planning process is conducted at skill level instead of task and skill levels. Given the specific application scenarios, this skill-level-centric planning is expected to generate a feasible and efficient workflow tailored to each production agent, including a sequence of task-and-skill pairs. As a result, tasks are precisely allocated to the most suitable agents, ensuring harmonious and efficient collaboration throughout the production process. This planning result not only enhances cooperation among production processes but ensures the full utilization of all production resources.

E. Consistency Checking and Ontology Reasoning

Planning outcomes are subsequently saved back into the ontology, triggering consistency checks and ontology updates, for instance, revising average execution times. These updates facilitate further analysis and optimization of robot skill programming parameters. Additionally, new knowledge, such as product variants, are deduced through inference rules, offering recommendations that are subsequently subjected to validation by engineers.

IV. Experiments and Results

Gear assembly, a product requiring multifaceted skills, serves as the foundation for this study. Two use cases were selected and analyzed to demonstrate the feasibility and flexibility of the proposed method.

A. Use Case Description

In this paper, we focus on the gear assembly to evaluate the effectiveness and feasibility of the developed method. We consider the following two use cases (UCs):

UC 1: Large-scale Assembly Production Planning and Scheduling: This use case focuses on the process scheduling for producing 3D-printed gear components [1] within a production line that employs specialized robots in a multi-robot system for high-volume manufacturing, as illustrated on the bottom left in Figure 3. We extended the gear model with including a screw in the base. The selection of a 3D-printed model over an actual industrial gearbox is intentional; our primary interest lies not in the precise execution of assembly tasks but in the broader aspects of planning and scheduling with multiple robots. Employing models can also ensure the method's generalizability and aid in potential future extensions.

UC 2: Small-scale Assembly Production Planning: This use case utilizes the Wittenstein NP planetary gearbox, series NPL 025S-MF2-70 [2], as a representative product, with components illustrated on the bottom right in Figure 3. The scenario caters to the specific needs of small-scale production, underscoring the variability in assembly steps arising from process complexity. By altering gearbox parameters, such as gear size based on customer requirements, this case demonstrates product customization, enabling straightforward setup for small to medium-sized factories.

B. System Setup and Requirements

In UC 1, coordination among different modules within the assembly process builds on our previous research [31]. The layout plan is depicted on the top left of Figure 3. This involves integrating a centralized planning module, Central.AI, with decentralized modules, including a supermarket cell with a Franka robot integrated on a moving platform (Dynamic Agile Production Robots That Learn and Optimize Knowledge And Operations - Darko), a mobile robot (MiR250), and assembly robots (Franka robots). Central.AI aggregates existing skills from these decentralized modules and undertakes the planning task sequence planning and scheduling, distributing tasks among decentralized robotic modules, which then perform the tasks according to the instructions from Central.AI.

In UC 2, depicted in Figure 3, an initial setup is required to accommodate the small batch production needs. As a "flexible re-configurable assembly station", this setup comprises a Franka robot, a mobile manipulation platform and a self-designed gripper tip plus a tool change set (4 tool-cubes) from Leverage Robotics. Adjustments of robots to perform existing skills and obtain the precise location of objects are executed and archived in a library for subsequent use. This arrangement allows for reconfiguration to adapt to various production demands.

The Central.AI system, the manipulation system of the MiR and robotic units, utilizes the Ubuntu operating system, requiring a minimized version of 20.02. Programming languages are specified for Python, necessitating a minimum version of 3.8. Ontology generation is conducted using the Ontor Python package[3], while reasoning is facilitated through the Owlready2 package[4]. Assembly operations utilize a set of self-developed taxonomy robotic manipulation skills [32], [33] equipped with a real-time kernel. Two robots have been integrated with the camera Realsense D435i. The MiR 250 mobile robot is controlled via HTTP requests, ensuring streamlined operations. Module communication is facilitated via WebSockets to ensure efficient data exchange. Task planning utilizes four PDDL planners within planutils environment [5]: POPF, Optic, SMTPlan and LPG-td, for providing different plan results to choose. Ontology reasoning and inference are taken using the Pellet Reasoner and Semantic Web Rule Language (SWRL) rules, enriching the method's analytical capabilities and modeling the ontology in Web Ontology Language (OWL).

C. An Overview of Robot Skills

Different robots and grippers entail varying skills across the two use cases, as detailed in Figure 4 and Table III. UC 1 employs specialized robots in a production line setting, each performing a distinct skill to the overall process. Conversely, UC 2 uses only one robot but with five grippers with each performing different assembly skills, enabling a versatile and adaptable assembly process.

[1] The gear assembly task in the Siemens Innovation Challenge: https://www.therobotreport.com/reinforcement-learning-industrial-robotics/

[2] https://alpha.wittenstein.de/en-en/products/np-planetary-gearbox/

[3] https://github.com/felixocker/ontor

[4] https://owlready2.readthedocs.io/en/latest/

[5] https://github.com/AI-Planning/planutils

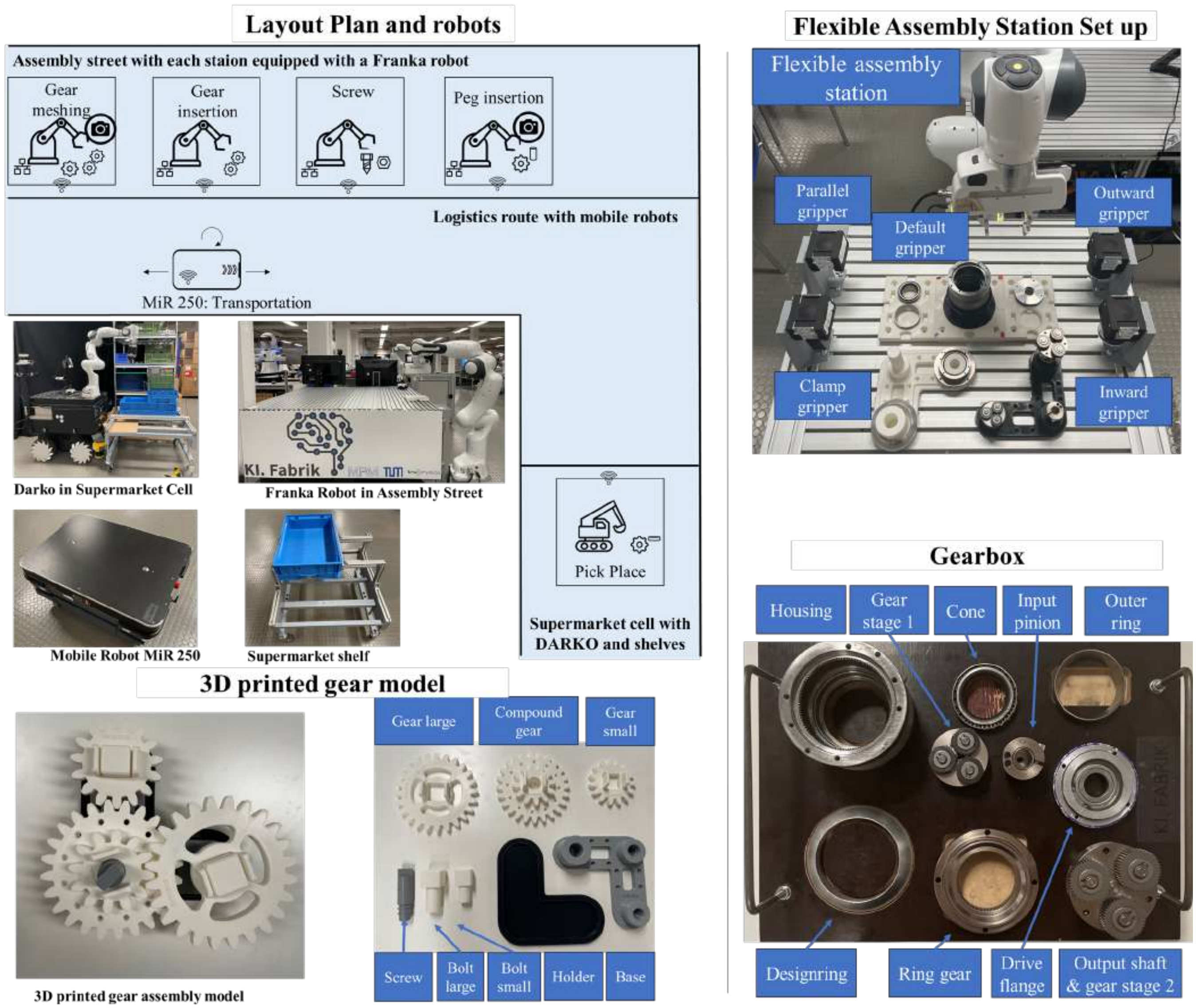

Fig. 3. Two use cases: UC1 (left) shows dynamic scheduling for 3D printed gear model assembly with multi-robots; UC 2 (right) shows planning for Wittenstein gearbox assembly with a single robot and multi-grippers.

TABLE III
GRIPPERS AND SUITED OBJECTS IN UC 2

Gripper	Strock (mm)	Object shape	Example
Default gripper	0-87	Uniform	Output shaft & gear stage 2, design ring, ring gear, drive flange, input pinion
Parallel gripper	33-60	Uniform, round	ring gear, Gear stage 1
Clamp gripper	75-120	Long, (ir-)regular	design ring
Outward gripper	16-61	Round inside	Cone, outer ring
Inward gripper	15-50	Round outside	Cone, outer ring, gear stage 1

D. Planning Results and Planning Execution

Since assembly processes differ due to variations in gear part relationships and the execution times of different robots/-grippers for various skills, the resulting plans are not unique. In UC 1. only LPG-td planner generated plans, as other planners exceeded the 60-second planning threshold. The optimised selected planning result for UC 1, depicted in Figure 5, outlines the sequence of process steps, indicating the specific tasks assigned to each robot at given times. For UC 2, five plans were generated and the best plan is shown in Figure 6. The selection of optimal planning outcomes relies on weighted criteria with $w_t = 0.6$, $w_s = 0.3$ and $w_R = 0.1$. Different requirements may have varying priorities. In this context, we emphasize execution time as the most critical factor for the assembly process, prioritizing it above the

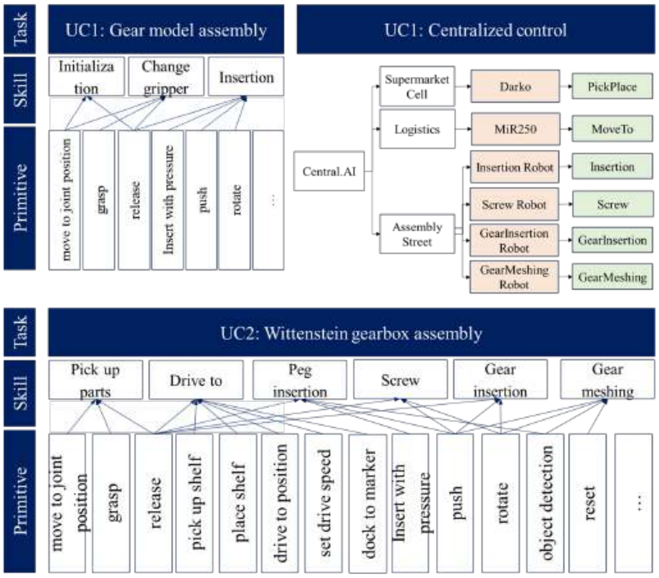

Fig. 4. An overview of robot skills and primitives

sequence of steps and resource utilization. This is adopted to align with the efficiency and timeliness goals of industrial production. With this weight setting, the LPG-td planner preferred for both use cases.

The task allocation for UC 1 employs a hierarchical structure, integrating a local planner with individual task queues designated for each assembly station. These queues, unique to each workstation, prioritize tasks awaiting execution. Under standard production conditions, queues comprise solely regular tasks. When the plan is generated for a special product, the production process is accordingly adjusted. Upon the MiR's arrival at a workstation and readiness for a specialized task, the local planner dispatches a request for this task. The workstation, upon acknowledging this request, incorporates the special task into its queue with a higher priority. This ensures immediate execution after the current task. Upon completing the special task, the workstation signals the local planner for the scheduling and execution of subsequent tasks. Then, the workstation resumes processing regular tasks in its queue, maintaining operational continuity. In UC 2, the robot executes the assembly task strictly according to the generated plan, ensuring adherence to predetermined workflows.

These selections are then integrated into the ontology for further analysis, utilizing the Pellet Reasoner and SPARQL for inconsistency checks, particularly when actual execution times deviate from historical averages for a given skill. For instance, the execution time for skills applied to the Wittenstein gearbox varies due to the lack of extensive historical data. Adjustments and updates are then applied to the ontology by averaging the execution time of historical data and new data. Furthermore, by applying inference rules tailored for identifying potential production enhancements, new product opportunities are identified, showcasing possible interrelations between components, e.g., a simplified gear product with only two bolts and two gears on base in UC 1. This process facilitates the identification of possible execution time improvements and suggests possibilities for new product development. An overview of the inconsistencies and new knowledge found are listed in Table IV.

TABLE IV
INCONSISTENCIES AND NEW KNOWLEDGE AFTER PLANNING

Reasoner and query output	Type	Examples
Inconsistencies	Execution mismatch	Skill deployment, execution time, gripper-object compatibility
	Renewal error	Unplanned gripper replacement
New Knowledge	Product variants	New design variants with component constraints
	Execution enhancement	Robot allocation within required time

E. *Evaluation of execution time*

The total execution time for procedural steps in both use cases is detailed in Table V. For UC1, no setup is required as the planning is based on the current setup, necessitating only the execution of the generated scheduling plan. Conversely, UC2 requires additional time for setup including printing holders for objects, mounting grippers and holders on the plate, adjusting robot skills through programming modifications. The total time required for planning is reasonable, supporting the feasibility of near-real-time execution for the proposed planning method. However, runtime ontology updating and reasoning remain unfeasible and need to be further investigated.

V. CONCLUSION AND OUTLOOK

In conclusion, this paper introduces an automatic method of ontology-based AI planning and scheduling in production processes, specifically tailored to gearbox assembly use cases. By utilizing AI planning techniques, multi-robot coordination, and knowledge analysis, the proposed method demonstrates dynamic, near real-time characteristics that significantly enhance efficiency, adaptability, and productivity within assembly line operations. The two use cases further illustrate the method's potential in scheduling and planning for both large and small production batch sizes, showcasing its applicability across various manufacturing processes.

Future work will focus on increasing the complexity of the use cases to highlight the capability of integrating various techniques, such as incorporating additional sensors for real-time environmental perception and feedback on production, and utilizing large language models and behavior trees for more efficient skill teaching, thus reducing setup time. Additionally, we plan to explore machine learning algorithms for improved multi-robot collaboration and knowledge reasoning. Further research will also delve into the robot skill optimization and product design through the use of inferences within ontologies.

Fig. 5. The planning result and execution of gear assembly in UC 1: picture in each column showcases the pre, executing and post robot state for each planned process

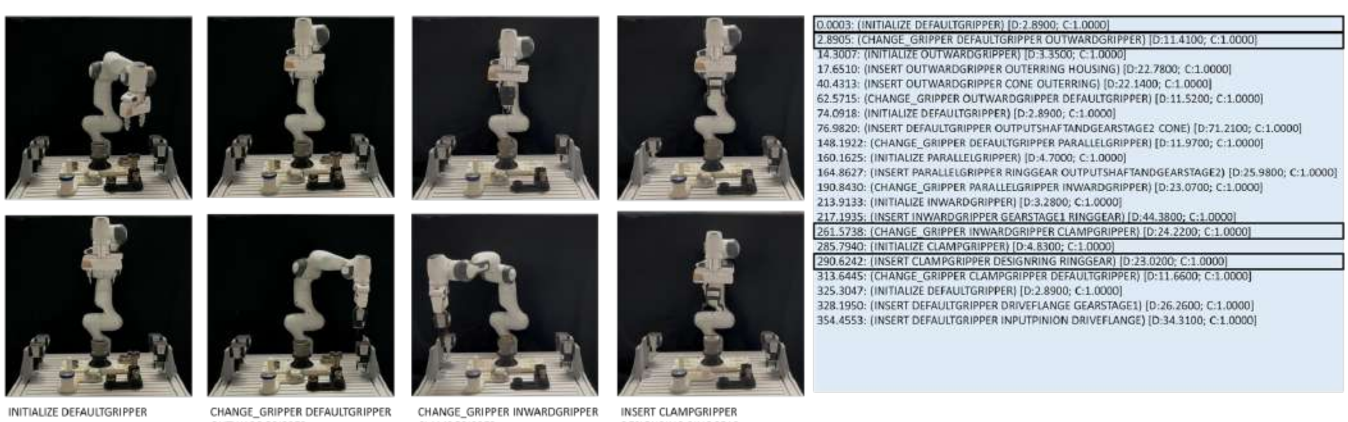

Fig. 6. The planning result and execution of Wittenstein gearbox assembly in UC2: picture in each column showcases the pre and post robot state for each planned process

TABLE V

ESTIMATED EXECUTION TIME FOR EACH STEP

UC	Setup					Online planning and task allocation				Offline analysis		
	Holder design	Holder printing	Plate mount.	Robot system config. and skills adjust.	Total	Planning	Plan select.	Task alloc.	Total	Consist. Check	Infer.	Total
1	-	-	-	-	-	6.35s	0.004s	0.0006s	6.3546s	213.90s	434.32s	648.22s
2	5h	45h	0.5h	48h	98.5h	5.32s	0.006s	0.0005	5.3265s	186.32s	231.32s	417.64s

ACKNOWLEDGMENT

The authors acknowledge the financial support by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK (Phase 1: Infrastructure as well as the research and development program under grant no. DIK0249). Please note that S. Haddadin had a potential conflict of interest as a shareholder of Franka Emika GmbH.

REFERENCES

[1] F. Ocker, C. J. Paredis, and B. Vogel-Heuser, "Applying knowledge bases to make factories smarter," *at - Automatisierungstechnik*, vol. 67, no. 6, pp. 504–517, 2019.

[2] S. Staab and R. Studer, *Handbook on ontologies*. Springer Science & Business Media, 2010.

[3] F. Van Harmelen, V. Lifschitz, and B. Porter, *Handbook of knowledge representation*. Elsevier, 2008.

[4] D. Long, J. Dolejsi, and M. Stolba, "Scheduling problems in pddl," in *Workshop on Knowledge Engineering for Planning and Scheduling*, 2023.

[5] I. Gocev, S. Grimm, and T. A. Runkler, "Explanation of action plans through ontologies," in *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, pp. 386–403, Springer International Publishing, 2018.

[6] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with dolce," in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web* (A. Gómez-Pérez and V. R. Benjamins, eds.), pp. 166–181, Springer Berlin Heidelberg, 2002.

[7] I. Niles and A. Pease, "Towards a standard upper ontology," pp. 2–9, 10 2001.

[8] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko, "Mason: A proposal for an ontology of manufacturing domain," vol. 2006, p. 195 – 200, 2006.

[9] A. Köcher, C. Hildebrandt, L. M. Vieira da Silva, and A. Fay, "A formal capability and skill model for use in plug and produce scenarios," in *IEEE ETFA*, vol. 1, pp. 1663–1670, 2020.

[10] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, *et al.*, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1193–1204, 2013.

[11] S. Balakirsky, Z. Kootbally, T. Kramer, A. Pietromartire, C. Schlenoff, and S. Gupta, "Knowledge driven robotics for kitting applications," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1205–1214, 2013. Ubiquitous Robotics.

[12] Z. Kootbally, C. Schlenoff, C. Lawler, T. Kramer, and S. Gupta, "Towards robust assembly with knowledge representation for the planning domain definition language (pddl)," *Robotics and Computer-Integrated Manufacturing*, vol. 33, pp. 42–55, 2015. Special Issue on Knowledge Driven Robotics and Manufacturing.

[13] E. Järvenpää, N. Siltala, O. Hylli, and M. Lanz, "The development of an ontology for describing the capabilities of manufacturing resources," *Journal of Intelligent Manufacturing*, vol. 30, 02 2019.

[14] N. Lohse, H. Hirani, S. Ratchev, and M. Turitto, "An ontology for the definition and validation of assembly processes for evolvable assembly systems," in *The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing, 2005.*, pp. 242–247, 2005.

[15] C. Schlenoff, C. Schlenoff, F. Tissot, J. Valois, and J. Lee, *The process specification language (PSL) overview and version 1.0 specification*. US Department of Commerce, NIST, 2000.

[16] M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden, S. Penberthy, D. Smith, Y. Sun, and D. Weld, "Pddl - the planning domain definition language," *Technical Report, Tech. Rep.*, 08 1998.

[17] M. Fox and D. Long, "Pddl2. 1: An extension to pddl for expressing temporal planning domains," *Journal of artificial intelligence research*, vol. 20, pp. 61–124, 2003.

[18] A. Rogalla, A. Fay, and O. Niggemann, "Improved domain modeling for realistic automated planning and scheduling in discrete manufacturing," in *2018 IEEE ETFA*, vol. 1, pp. 464–471, 2018.

[19] B. Wally, J. Vyskocil, P. Novak, C. Huemer, R. Sindelar, P. Kadera, A. Mazak, and M. Wimmer, "Production planning with iec 62264 and pddl," vol. 2019-July, p. 492 – 499, 2019.

[20] G. Wickler, L. Chrpa, and T. L. McCluskey, "Kewi: A knowledge engineering tool for modelling ai planning tasks," *KEOD*, p. 36 – 47, 2014.

[21] G. Papadimitriou and D. Lane, "Semantic based knowledge representation and adaptive mission planning for mcm missions using auvs," in *OCEANS 2014 - TAIPEI*, pp. 1–8, 2014.

[22] R. Bertolucci, A. Capitanelli, C. Dodaro, M. Maratea, F. Mastrogiovanni, and M. Vallati, "Kr&r approaches for robot manipulation tasks with articulated objects," in *Proceedings of the RiCeRcA Workshop*, vol. 2272 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018.

[23] P. J. S. Gonçalves, J. R. Caldas Pinto, and F. Torres, "Knowledge and capabilities representation for visually guided robotic bin picking," in *Robot 2019: Fourth Iberian Robotics Conference*, pp. 429–440, Springer International Publishing, 2020.

[24] O. Ruiz, J. Rosell, and M. Diab, "Reasoning and state monitoring for the robust execution of robotic manipulation tasks," in *IEEE ETFA*, pp. 1–4, 2022.

[25] S. Joo, S. Bae, J. Choi, H. Park, S. Lee, S. You, T. Uhm, J. Moon, and T. Kuc, "A flexible semantic ontological model framework and its application to robotic navigation in large dynamic environments," *Electronics*, vol. 11, no. 15, 2022.

[26] O. Ruiz-Celada, A. Dalmases, R. Suárez, and J. Rosell, "Be-aware: an ontology-based adaptive robotic manipulation framework," in *2023 IEEE ETFA*, pp. 1–4, 2023.

[27] T. Hoebert, W. Lepuschitz, M. Vincze, and M. Merdan, "Knowledge-driven framework for industrial robotic systems," *Journal of Intelligent Manufacturing*, vol. 34, no. 2, p. 771 – 788, 2023.

[28] T. Hoebert, D. Neubauer, M. Merdan, W. Lepuschitz, S. Thalhammer, and M. Vincze, "Ros-driven disassembly planning framework incorporating screw detection," in *IEEE CCE*, pp. 1–6, 2023.

[29] M. Mayr, F. Rovida, and V. Krueger, "Skiros2: A skill-based robot control platform for ros," 2023.

[30] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, 2016.

[31] B. Vogel-Heuser, Y. Deshpande, F. Bi, J. Zhao, D. Hujo, W. Kellerer, A. Kraft, B. Vojanec, and T. Markert, "Delay modelling and measurement of multi-agent systems with digital twins in a gear assembly use case," in *IEEE CASE*, pp. 1–8, 2023.

[32] Y. Wu, F. Wu, L. Chen, K. Chen, S. Schneider, L. Johannsmeier, Z. Bing, F. Abu-Dakka, A. Knoll, and S. Haddadin, "1 khz behavior tree for self-adaptable tactile insertion,"

[33] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *ICRA*, pp. 5844–5850, IEEE, 2019.