Using Ontologies for Adaptive Planning and Robust Execution in Robotic Manipulation

Oriol Ruiz-Celada[1], Víctor Molina[1], Raúl Suárez[1], Muhayy Ud Din[2], Isiah Zaplana[1] and Jan Rosell[1,*]

[1]Inst. of Industrial and Control Engineering (IOC), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[2]Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University, United Arab Emirates.

Abstract

Robotic manipulation tasks face many challenges when dealing with real scenarios that require the perception and understanding of the environment, planning with procedures that can adapt to the actual situation, and monitoring and recovery capabilities to ensure successful execution despite uncertainties and errors. The use of knowledge in terms of ontologies can address these needs by providing reasoning capabilities that allow the robot to be aware of the situation and the task and to dynamically create execution structures that adapt its behaviors in a flexible and robust manner. In this line, this paper discusses the work done at the Institute of Industrial and Control Engineering (IOC-UPC) that uses ontologies combined with Large Language Models to plan and automatically generate Behavior Trees with monitoring and recovery capabilities, which allow for the control of robot execution for successful performance in real scenarios.

Keywords

Robotics, Manipulation planning, Execution configuration, Ontologies, Large Language Models

1. Introduction

Robotic manipulation tasks in real-world scenarios, such as those faced by a dual-arm mobile manipulator performing assistive tasks in a human environment, pose significant challenges due to uncertainty and their dynamic nature. This demands intelligent systems capable of adapting planning and execution to unforeseen changes. That is, the robot needs to be aware of the situation, of its own capabilities, of the task particularities and of the goal state to be achieved, which will condition the planning. On the other hand, to execute the plan the robot needs to be able to flexibly configure execution structures that also account for monitoring actions and recovery strategies, in order to ensure a robust performance of the task.

Several tools can be used to achieve this objective, such as automated planning, Knowledge Representation and Reasoning (KR&R), Large Language Models (LLMs) and Behavior Trees (BTs). Automated planning [1] (also referred as task planning in the robotics context) focuses on the generation of the action sequences that may allow a robot to perform a given task. It is based on symbolic models of actions and states, usually using the Planning Domain Definition Language (PDDL [2]). Task planning can be enhanced with Knowledge Representation and Reasoning (KR&R [3]), to facilitate the modeling of the world, the interpretation of situations, and to make informed decisions without hard-coded rules, i.e., to support an intelligent and adaptive behavior. KR&R is usually based on ontologies, which provide a structured and semantic representation of domain knowledge that allows reasoning about the environment and the task, facilitating the robot's adaptation of the plan to the current circumstances. A broad overview of ontology-based autonomy in robotics is provided in [4]. As an example, the CRAM framework [5] has been conceived to empower service robots with cognitive capabilities for

Proceedings of the Joint Ontology Workshops (JOWO) - Episode XI: The Sicilian Summer under the Etna, co-located with the 15th International Conference on Formal Ontology in Information Systems (FOIS 2025), September 8-9, 2025, Catania, Italy
*Corresponding author.
oriol.ruiz.celada@upc.edu (O. Ruiz-Celada); victor.molina.diez@upc.edu (V. Molina); raul.suarez@upc.edu (R. Suárez); muhayyuddin.ahmed@ku.ac.ae (M. U. Din); isiah.zaplana@upc.edu (I. Zaplana); jan.rosell@upc.edu (J. Rosell)
0000-0002-6171-7270 (O. Ruiz-Celada); 0000-0002-3853-7095 (R. Suárez); 0000-0002-3853-7095 (M. U. Din); 0000-0002-0862-3240 (I. Zaplana); 0000-0003-4854-2370 (J. Rosell)

everyday manipulation tasks, while RoboPlanner [6] supports dynamic decision-making in industrial environments by combining semantic models with autonomous planning. LLMs are advanced AI systems trained on massive amounts of text data. They can interact with human language in a coherent and context-aware manner, enabling them to flexibly enhance task planning and facilitate human-in-the-loop systems where humans actively collaborate with robots during task execution. Finally, BTs are execution structures formed by directed rooted trees with different types of nodes: (inner) control nodes used to control the execution flow of their children and (leaf) execution nodes that represent single functionalities. BTs are usually used in robotics to define skills for their modular and reactive capabilities.

In this paper, we discuss the use of these tools in pursuing autonomy in robots performing manipulation tasks in the real world. Section 2 reviews the proposal of a knowledge-based manipulation framework. Section 3 reviews planning approaches that combine ontologies and LLMs. Section 4 reviews the automatic execution configuration of BTs and proposes new BT structures to add monitoring and recovery strategies flexibly. Finally, Section 5 presents the conclusions and discusses the future research directions.

2. The BE-AWARE Framework

The BE-AWARE framework [7] is a robotics manipulation framework, proposed by the authors, that integrates situation, domain, and execution awareness using ontologies, focusing on adaptability through ontological reasoning in order to plan and execute manipulation tasks.

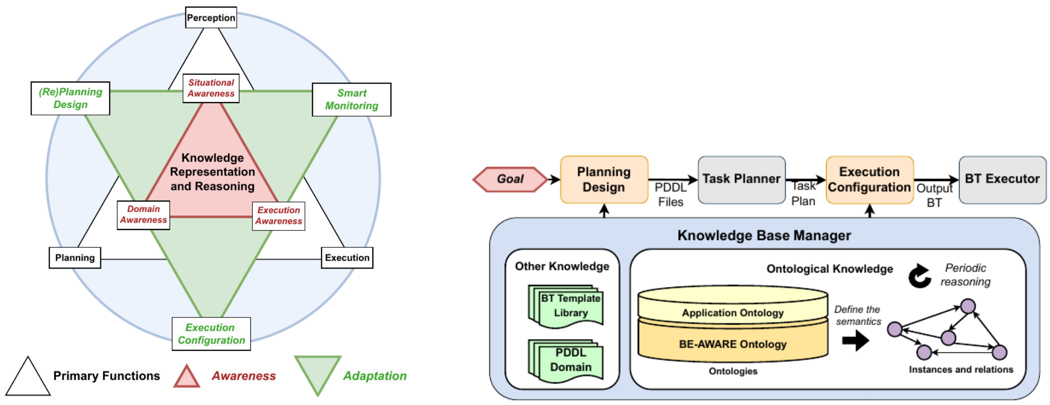


Figure 1: The Be-Aware framework: (left, taken from [7]) illustrates the three primary functions of a robotic system (perception-planning-action) enhanced with knowledge through the awareness modules, which are also used by the adaptation modules to give adaptation capabilities to the robot at planning, execution and monitoring levels; (right, taken from [8]) details the use of ontologies by two of the adaptation modules: the planning design to tailor the PDDL files and the execution configuration to tailor the BT execution structures.

Fig. 1 (left) shows the main parts of the framework: (1) the *Knowledge Representation and Reasoning core* (BE-AWARE contains Situation, Domain and Execution ontologies, which are built upon the Autonomous Robotics Ontology, AuR [9], a IEEE standard that contains common terms for autonomous robots); (2) *Awareness modules* that enhance the primary functions of perception, planning and action (situation awareness is built on relations between entities that allow to understand the scene, domain awareness arises from the knowledge of the predicates that define the system state and the actions that allow its changes, and execution awareness is based on knowledge of the execution structures); (3) *Adaptation modules* that give adaptation capabilities at planning level (automatically generate the PDDL problem file), at execution level (automatically generate the BTs for task execution) and at monitoring level (automatically generate the monitoring and recovery BT structures).

Fig. 1 (right) shows the planning and execution pipeline, where the ontologies play a role in the planning design and in configuring the BTs for the execution, detailed below in Sections 3.1 and 4.1, respectively.

3. Planning with Ontologies and LLMs

3.1. Situation and Task Awareness

A situation ontology is included in the BE-AWARE framework in order to help the robot to understand the scene, i.e. to have situation awareness. The ontology includes concepts for describing the relative location between objects, the ontology instances being provided by a smart perception module [10] designed to continuously perceive the environment (i.e., detect objects and their poses) and to periodically perform geometric reasoning to determine the relative positions (blue blocks in Fig. 2). To address ambiguity and incompleteness in perception, BE-AWARE also integrates LLMs to populate ontologies from natural language user inputs [11] (orange blocks in Fig. 2). The populated ontologies are then processed using OWL-based reasoning tools, enabling robust world modeling.

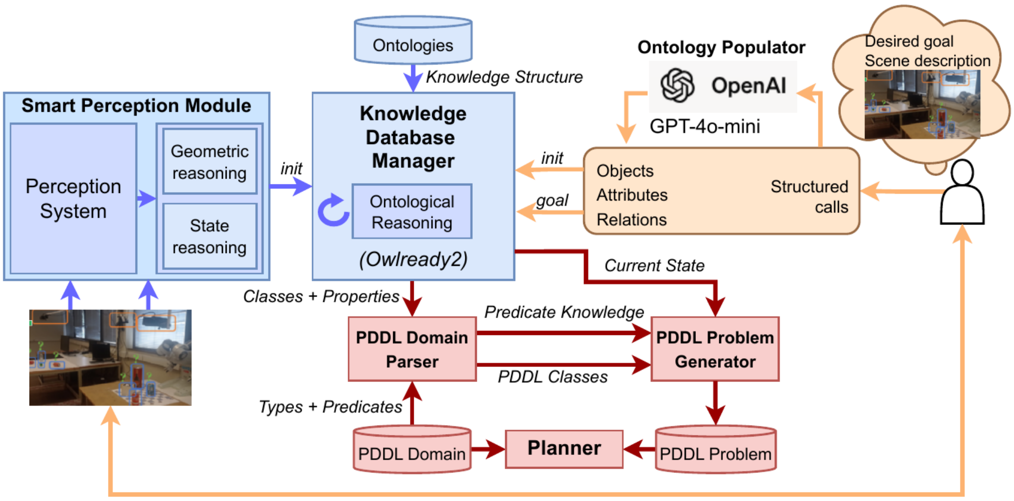


Figure 2: Knowledge-based framework with LLM ontology population and PDDL generation (taken from [11]).

The LLM instantiation feature is also exploited by BE-AWARE to enable the user to define the goal state of a manipulation task. BE-AWARE assumes that the PDDL domain is fixed and given, and it uses this domain, together with the situation ontology, to generate the PDDL problem file. This is done by matching ontological concepts to PDDL types and predicates, resulting in a problem file that includes the PDDL objects, the initial state, and the goal state (red blocks in Fig.2). In this way, BE-AWARE equips the robot with the task awareness necessary to solve the task using classical task planners [1].

3.2. Ontology-driven Prompt Tuning

As an alternative to classical task planning, LLM-based approaches have been proposed to exploit the potential of LLMs, although the resulting plans have not always been accurate enough (with flaws such as incorrect temporal goal ordering) due to static and template-based prompting, which has problems in adapting to dynamic environments and complex task contexts. Following this line, and complementary to the proposal described in the previous section, we explored LLM-based task planning, but addressing its limitations with a novel ontology-driven prompt-tuning framework that employs knowledge-based reasoning to refine and expand user prompts with task contextual reasoning and knowledge-based environment state descriptions. The integration of domain-specific knowledge into the prompt ensure semantically accurate and context-aware task plans. As illustrated in Fig. 3, the elaborated prompt lead the LLM to resolve errors such as incorrect object placement order (e.g., placing food before crockery), improving the plan's semantic integrity.

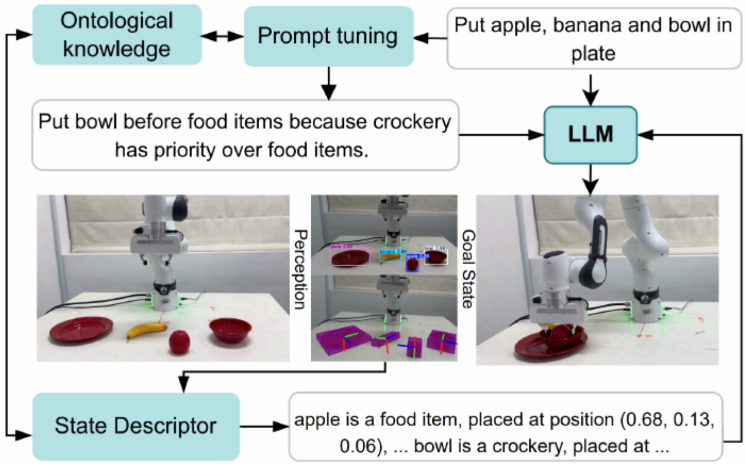


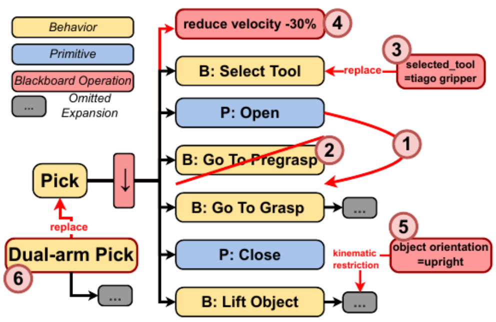
Figure 3: The ontology-driven LLM-TAMP framework (taken from [12]).

4. Adaptive and robust execution configuration

In order to execute the actions of a task plan, like Pick, Move, Place, an execution structure, such as BTs, is required. We have proposed a flexible, ontology-based automatic generation of BTs (without using LLMs), tailored to the current circumstances and specific requirements of a task, that encapsulate execution details, many of which require information not available at plan time.

4.1. Adaptive Behavior Tree generation

BTs provide modularity and reactivity that are ideal for robot execution, but, traditionally template-based BTs are used, which prevent the robot to adapt to the current circumstances if they have not been considered in the BT. To go beyond this limitation, the work in [8] proposed the concept of *Flavors*, as formalized modifiers that, based on reasoned contextual knowledge and starting from a basic BT template, tailors it by, for instance, reordering nodes, changing parameters, adding kinematic constraints, or skipping/replacing sub-tasks depending on the properties of the involved entities. As an example, Fig. 4 illustrates multiple Flavors applied to a generic Pick template. Flavor 2 eliminates the go-to-pregrasp sub-task when the object is considered small, while Flavor 4 adds a velocity parameter to reduce the speed if the object is considered dangerous. The properties that condition these Flavors are derived from the reasoned ontology.



Flavor Description
1.If the robot has a hand, **Delay** `Open` until after `GoToPregrasp`
2.If the object is small, **Ignore** `GoToPregrasp`
3.If the robot only has one possible tool, **Replace** `SelectTool` by directly setting the tool in the BT
4.If the object is dangerous, **Reduce** velocity by 30%
5.If the object is a glass with liquid, add the **Kinematic restriction** to be kept upright
6.If the object is heavy, **Replace** `Pick` by `Dual-arm Pick`

Figure 4: Flavored BT generation (taken from [8]).

4.2. Extensions for monitoring and recovery

To enhance the execution layer with robustness and reactivity, we propose here the use of Behavior Blocks, modular BT substructures tailored for monitoring and recovery. These blocks are instantiated from templates and further customized using ontologically driven flavors, in a similar way as done in the previous section. Each monitored behavior is wrapped with conditions that assess the state of the system at three points: preconditions (before execution), hold conditions (during execution), and postconditions (after execution), as shown in Fig. 5. Upon failure, a diagnostic code is generated, triggering a recovery strategy. The basic template includes retries (executing the behavior up to N times), but more sophisticated strategies can be selected based on the failure type, such as motion replanning, alternative action selection, or environmental interaction (e.g., obstacle removal).

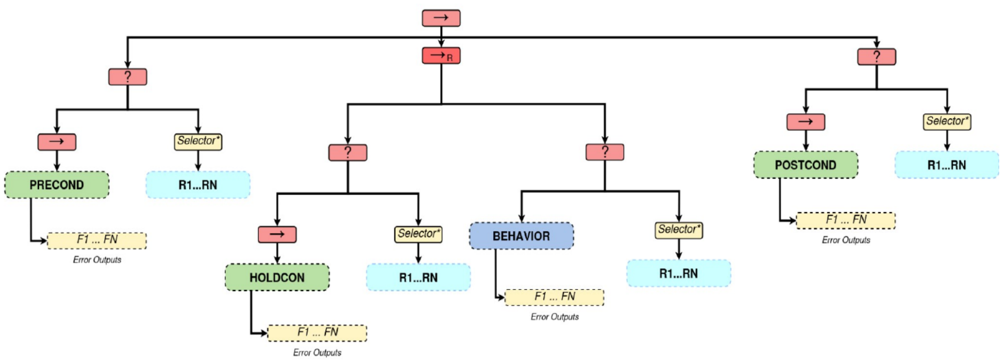


Figure 5: Behavior Block template for monitoring and recovery.

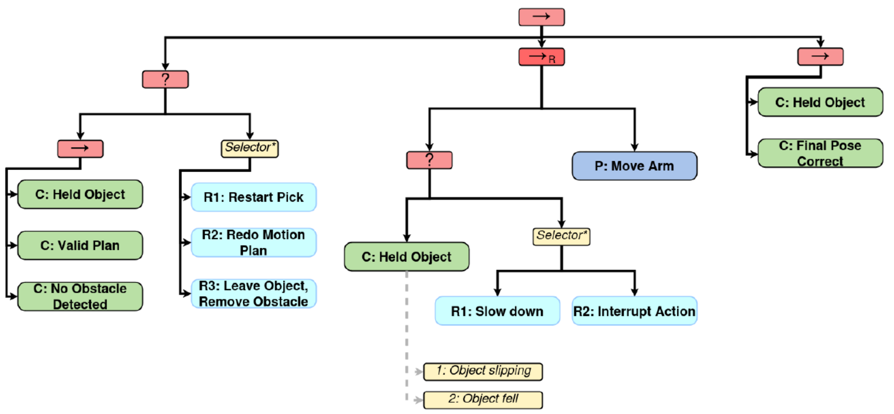


Figure 6: Behavior Block for monitoring and recovery applied to the *Move Arm* action.

The instantiation and tuning of monitoring and recovery structures is driven by semantic reasoning over the ontologies, which takes into account factors such as sensor availability or object properties. For instance, if the object is a fragile glass containing liquid, the system may insert an orientation check in the monitoring phase (the function implementing this check will determine the failure occurrence and generate a diagnostic code); if a slippery or hazardous item is detected, the system can tighten trajectory tolerances or reduce motion speed proactively. Fig. 6 illustrates the Behavior Block used for monitoring and recovery in the *Move Arm* action.

5. Conclusions and Future Work

The aim of this paper has been to illustrate the potential of using ontologies to enhance both planning and execution in robotic manipulation, in particular, making planning adaptive to the circumstances that the robot encounters and making execution robust to uncertainties, thus contributing to make robots truly autonomous. A unified framework has been conceived with several ontologies at its core, which allowed: (a) the context-aware instantiation of symbolic problem definitions; (b) the tuning of prompts for LLM-based task planning; (c) the generation of optimized execution structures tailored to the current situation and task; and, (d) the generation of monitoring and recovery execution structures, based on the actions to be performed and the features and circumstances of the actual execution.

Future work includes the expansion of the Flavor library to support a wider range of behavioral adaptations, the improvement of failure handling through better integration of sensory context and reasoning mechanisms, and the testing on real robots performing everyday tasks in human environments.

Acknowledgments

This work was partially supported by the European Commission's Horizon Europe Framework Programme with the project IntelliMan (AI-Powered Manipulation System for Advanced Robotic Service, Manufacturing, and Prosthetics) under Grant Agreement 101070136.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT to improve the writing style and for grammar and spelling checks. After using these tool(s)/service(s), the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Ghallab, D. Nau, P. Traverso, Automated planning: theory & practice, Elsevier, 2004.
[2] M. Ghallab, A. Howe, C. Knoblock, D. Mcdermott, A. Ram, M. Veloso, D. Weld, D. Wilkins, PDDL—The Planning Domain Definition Language, Technical Report, The AIPS-98 Planning Competition Committee, 1998.
[3] M. Beetz, Knowledge representation and reasoning, in: A. Cangelosi, M. Asada (Eds.), Cognitive Robotics, MIT Press, 2022, pp. 413–432.
[4] A. Olivares-Alarcos, D. Beßler, A. Khamis, P. Goncalves, M. Habib, J. Bermejo-Alonso, M. Barreto, M. Diab, J. Rosell, J. Quintas, J. Olszewska, H. Nakawala, E. Pignaton, A. Gyrard, S. Borgo, G. Alenyà, M. Beetz, H. Li, A review and comparison of ontology-based approaches to robot autonomy, The Knowledge Engineering Review 34 (2019) 1–29.
[5] M. Beetz, G. Kazhoyan, D. Vernon, S. Profanter, The CRAM Cognitive Architecture for Robot Manipulation in Everyday Activities, 2023. arXiv:2304.14119.
[6] A. Kattepur, B. Purushotaman, Roboplanner: a pragmatic task planning framework for autonomous robots, Cognitive Computation and Systems 2 (2020) 12–22.
[7] O. Ruiz-Celada, A. Dalmases, R. Suárez, J. Rosell, BE-AWARE: an ontology-based adaptive robotic manipulation framework, in: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), 2023, pp. 1–4.
[8] O. Ruiz-Celada, J. Rosell, R. Suárez, Knowledge-based execution configuration of behavior trees, in: International Workshop on Ontologies and Standards for Robotics and Automation (WOSRA 2024) as part of the 2024 IEEE International Conference on Robotics and Automation, 2024.
[9] IEEE Standard for Autonomous Robotics (AuR) Ontology, IEEE Std 1872.2-2021 (2022) 1–49.
[10] O. Ruiz-Celada, A. Dalmases, I. Zaplana, J. Rosell, Smart perception for situation awareness in robotic manipulation tasks, IEEE Access 12 (2024) 53974–53985.
[11] V. Molina, O. Ruiz-Celada, R. Suárez, J. Rosell, I. Zaplana, Robot situation and task awareness using large language models and ontologies, in: Proceedings of the 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2025, pp. 96–103.
[12] M. Ud Din, J. Rosell, W. Akram, I. Zaplana, M. A. Roa, L. Seneviratne, I. Hussain, Ontology-driven prompt tuning for LLM-based task and motion planning, 2024. arXiv:2412.07493.