BE-AWARE: an ontology-based adaptive robotic manipulation framework

Oriol Ruiz-Celada
Inst. of Industrial and Control Eng.
Universitat Politècnica de Catalunya
Barcelona, Spain
oriol.ruiz.celada@upc.edu

Albert Dalmases
Inst. of Industrial and Control Eng.
Universitat Politècnica de Catalunya
Barcelona, Spain
albert.dalmases@estudiantat.upc.edu

Raúl Suárez
Inst. of Industrial and Control Eng.
Universitat Politècnica de Catalunya
Barcelona, Spain
raul.suarez@upc.edu

Jan Rosell
Inst. of Industrial and Control Eng.
Universitat Politècnica de Catalunya
Barcelona, Spain
jan.rosell@upc.edu

***Abstract*—Autonomous service robots are conceived to work in semi-structured and complex human environments performing a wide range of tasks and, hence, one of their main challenges is to be able to adapt the stages of the perceive-plan-execute cycle to perturbations ranging from small deviations on the poses of objects to large unexpected changes in the environment, as well as to recover from potential failures. To advance in this direction, this paper proposes an ontology-based manipulation framework where reasoning is used to enhance perception with situation awareness, planning with domain awareness and execution with the awareness of the execution structures. The combination of these different types of awareness allows the robot to have different adaptation capabilities. The conceptual schema of the framework is presented and discussed and the main future implementation challenges are pointed out.**

***Index Terms*—Robotic manipulation, knowledge, reasoning, ontology.**

I. Introduction and overview

Robotics is moving towards autonomy in order to be able to operate effectively in unstructured and complex environments [1]. An autonomous robot should be capable of sensing its environment, planning, and acting towards a specific goal without external control [2]. In order to endow the robots with this autonomy, they require intelligence, or at least a simulation of such. Thus, the field of Artificial Intelligence has grown with a wide variety of approaches that allow a robot to simulate intelligent behavior [3]. One of the fields of growing popularity is Machine Learning (ML). There are many applications of ML in robotics, from learning by demonstration, in which robots learn to perform a task by observing a human, to the use of neural networks to detect objects and their poses. Given input data, ML algorithms can recognize patterns within the data and extrapolate them. ML approaches, however, focus on a black-box approach to intelligence. They lack of explainability, i.e. the learned knowledge cannot be interpreted by a human [4], and makes it difficult to understand the decisions taken by the robot. Lack of transparency and explainability will affect the trustworthiness that autonomous robots require. In contrast, there is the subfield of Knowledge Representation and Reasoning (KR&R) [5], that represents knowledge in a manner that is understandable for both human and machine, so that the robot is able to comprehend the environment and derive information from such knowledge, achieving awareness. We will refer to Awareness as the concept of *knowing* information about the world and performing reasoning to *comprehend* it. In the framework, his awareness is divided in three levels:

- *Situational Awareness*: allows to reason on the state of the environment, the objects, the robots, their features and spatial relations.
- *Domain Awareness*: allows to reason on the planning domain actions and skills that the robots can perform.
- *Execution Awareness*: allows to reason on how the plans are to be executed by the robot and the state of the execution.

Nevertheless, awareness alone is not enough for a robot to be truly autonomous, as it also needs to exploit it to adapt to the changes in its environment. Adaptation is defined as "an event in which one (or more) robot, due to its evaluation of the current or expected future state, changes its current plan while executing it, into a new plan, in order to continuously pursue the achievement of the plan's goal" [6]. These changes range from slightly adjusting the movement at a geometric level to grasp an object properly, to having a full scale replan of the whole mission. With this in mind, this paper aims to contribute with the proposal of a framework for robotic manipulation with a knowledge-holistic view of the process, based on previous works related to perception [7] and planning [8]. The use of knowledge in all the stages of the process, in an integrated way, is envisaged to allow a fully smart and adaptive behavior of the robot.

In this paper we propose the BE-AWARE framework for robotic manipulation. The conceptual schema is shown in Fig. 1, where the triplet of Primary Functions (white triangle) representing the core functions that any basic manipulation framework needs to perform (perceiving the environment, planning towards a goal and executing the plan), is improved by two other triplets to make it smart and adaptive. The two contributions of the proposal are, first, the enhancement of each of the Primary Functions with the central Knowledge, resulting in the **Awareness** triplet (red triangle) that makes the framework smart, and second, the combination between the Awareness functions to form the **Adaptation** triplet (green triangle), that makes it adaptive.

This work was 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.

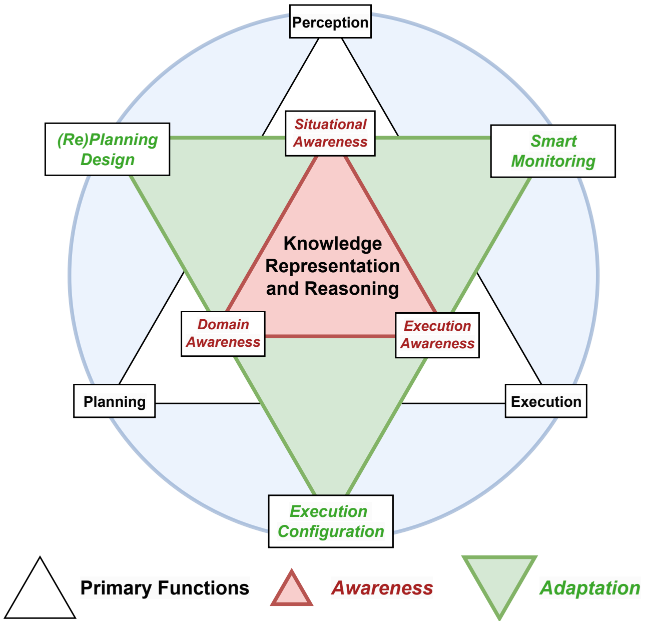

Fig. 1. Functional Schema of the framework

After this introduction, a review of related works is presented in Section II, then the Awareness and Adaptation are introduced, respectively, in Section III and IV. Finally, Section V discusses the contributions and future challenges.

II. Previous Works

There are other approaches with similar goals, the two most relevant being CRAM [9] and RoboPlanner [10]. CRAM is an innovative cognitive architecture designed for robots, to successfully carry out various everyday manipulation tasks. It allows for decision-making rooted in knowledge and reasoning over previous experiences, reasoning with motion and sensor data, as well as providing explanations for actions and their outcomes. CRAM uses KnowRob [11] as the central KR&R framework, which allows for situational awareness thanks to its simulation-based physics engine, having a digital twin of the environment and being able to "reason with its eyes and hands". However, it does not feature adaptation capabilities and does not have plan monitoring and failure handling, although it is indicated as a future development.

On the other hand, RoboPlanner is a framework that combines automated planning, execution monitoring, and adaptive deployments to enhance the cognitive autonomy of robots in uncertain scenarios. It allows robots to plan, execute, adapt, and re-plan while synchronizing with a knowledge base, enabling efficient and flexible robotic automation. While connected to a knowledge base, its reasoning capabilities are more limited than those of CRAM and of the approach proposed here, and does not feature the same level of awareness.

The core of all these approaches is the reasoning over the knowledge. For this, the use of ontologies arises as the most common tool for representing the semantics of knowledge in the field of robotics. In computer science, the concept of ontology defines a set of representational primitives that formalize a domain of discourse [12]. The Ontology Web Language (OWL) enables sharing these formalizations across applications. Standard ontologies have been created to unify the basic concepts applicable to a domain so that researches have a common base. The Autonomous Robotics Ontology (AuR) [13] contains common terms for autonomous robots and is the standard on which BE-AWARE is based.

Ontologies have been used in a variety of applications to enhance the Primary Functions. For Perception, the formalization in the ontology of the objects and the potential spatial relations between them may allow, once the sensor information is grounded, to reason on the object features and on the scene. For instance, different spatial relationships can be defined as symmetric or transitive in order to describe a setup of objects [14] and update and propagate these relationships constantly by the perception module.

For Planning, the existence of planning formalisms like the Planning Domain Description Language (PDDL) facilitates the description of actions, their pre-conditions and effects within the ontology, and allows to reason on the planning domain, like in the PMK framework [7] that uses Description Logic inference on the ontology to analyze runtime information, evaluate robot capabilities, constraints, and feasibility to perform Task and Motion Planning. Other approaches convert the information in the knowledge base and the ontology into PDDL configuration files to perform planning [15].

For Execution, plans can be automatically converted into execution procedures by querying the ontology, like the Knowledge Integration Framework (KIF) [16] that generates a State Machine for the skills that are described in its ontology.

For an extensive survey on existing ontology-based approaches to robot autonomy, the reader is referred to [17].

III. Awareness

The first contribution of the proposed framework is the Knowledge Base (KB) and the Awareness Functions. The Knowledge Base is formed by three ontologies, the Situation Ontology, the Domain Ontology, and the Execution Ontology, that share some concepts (e.g. the concept of *Robot*), and that are used, respectively, by the Situational Awareness Function, the Domain Awareness Function and the Execution Awareness Function.

A. *Situational Awareness*

Situational awareness is defined as the cognizance of objects in the world and understanding their meaning and the relationships with one another [18]. This definition is expanded to also include the state of the robots, such as the pose of the end effector and the joint states, as part of the situation. The raw data obtained from perception are largely geometric, such as the poses of the objects. A Smart Perception module handles the processing of these data and grounds them to symbolic properties and relations between entities following the Situation Ontology. For instance, data from the sensors may be received, giving the pose of a bottle and the pose of a table, and the grounding process may compare the poses and conclude that the bottle is *OnTopOf* the table. Moreover, with reasoning, the system can then infer that if the table is *In* the kitchen, then therefore the bottle is also *In* the kitchen. By repeating this process in every perception loop, the system is aware of all the entities in the world and their relations, and the state of the world is known.

B. Domain Awareness

The planning domain describes the laws and characteristics of a planning problem, marking the restrictions and the characteristics under which planning takes place. Robot skills, such as *pick up object* modify the state of the real world and change the state variables [19]. They are formed by a combination of the basic commands that the robot executes, called primitives, such as *close gripper*. To perform planning, symbolic planners require a domain that includes the skills and primitives that can be performed by the robots and the causal laws relating them to their preconditions and effects. Common sense knowledge of the domain is also included, tying the skills to object properties. In this way, the skill *open door* will be connected only to entities with doors, such as drawers.

Formalizing the domain in the Domain Ontology allows for richer descriptions of the primitives and skills and how they are related to the other elements in the domain. The information in the ontology is then written into the appropriate configuration files to be used by any PDDL-based planner.

C. Execution Awareness

At the lowest level of control, the execution is done through robot commands. A step above it, there are control architectures, such as Behavior Trees (BTs) [20], which control the flow of executions of the robot and the transitions between states of the system. The modular and reactive properties of BTs made them ideal for the framework. With the Execution Ontology, our contribution is to formalize how the planned skills and primitives are to be converted into an executable BT. Additionally, awareness of the control architecture allows for awareness of the state of the execution once it begins, understanding the progress being made on the plan, and comprehending the current and expected future state of the execution, information which can be used to enhance decision-making.

IV. Adaptation

The second contribution of the proposed framework is the proposal of how to exploit the Awareness to enhance the adaptive capabilities of the robot. The Adaptation triplet is formed by three main functions that allow the robot to leverage its awareness to keep working towards the goal despite the disturbances, i.e. the changes in the environment (like new or missing objects), the deviations (like differences in the expected object poses), or the failures (like non-desired effects of actions).

A. (Re)Planning Design

By combining the knowledge of the domain, the current situation of the world, and a desired goal, the problem is fully configured and planning can take place. Awareness of the situation allows the designed plans to be filled with a richer view of the state of the world than simple raw data, and awareness of the domain allows the designed plans to have a deeper description of the domain. It also allows to reason on which elements are relevant in order to be considered when planning a particular problem. For instance, if the current problem involves only manipulating objects on a table equipped only with a non-mobile robot, there is no need, initially, to include other objects and robots in the planning design process. This helps to reduce the planning computation time.

This process can be done offline to solve a particular planning problem. However, the resulting plan might be vulnerable to failures and deviations, and the framework needs to be capable of adapting to them. This is why, when adaptation is triggered, this function should be able to update the planning design with the current perceived situation, and perform replanning to amend the original nominal plan to find an alternative to advance towards the goal.

B. Execution Configuration

After a new plan has been obtained, either at the start or as a result of replanning, it must be translated into an appropriate BT structure for its execution, consisting of a sequence of BTs, each one corresponding to a skill of the solution plan. In static execution contexts the skill BTs can be configured from templates. However, having rigid templates of BTs lack the flexibility and generality required for adaptation in dynamic open environments. The challenge is how to use reasoning to obtain different BTs. For instance, the BT representing a “Pick” skill could be configured in many ways, and the modular nature of BTs allows for this configuration to be done in different stages. The initial stage involves configuring the functional properties of the skill (referring to those properties that have an effect in the world), including determining their parameters and organization as nodes inside the BT (e.g., picking a book may require to push it towards the edge of the table before it can be picked up). Then, the skill BT can be expanded and refined by adding the appropriate condition nodes necessary to successfully perform the skill, and recovery branches to correct the execution in case of failure by triggering a replan. For example, if an object falls from the gripper, the robot should pick it up before proceeding with the rest of the plan.

C. Smart Monitoring

Detecting when to adapt is key. The Execution Awareness allows the system to know, at any point of the execution, the Expected State, i.e. the state in which the system *should* be if everything so far has gone without issues. The Situational Awareness, on the other hand, contains the Observed State, i.e. the state in which the system *is*. If the Observed State and the Expected State differ, something has gone wrong and adaptation may need to take place. Monitoring needs to be fast in order to minimize the reaction time to these deviations. The Frame Problem [21] is the challenge of determining what new information introduced in the system is relevant or not, avoiding having to represent a significant number of implicitly obvious non-effects. Reasoning can be done to execute monitoring capabilities only on the relevant points of the execution, allowing for the ability to quickly monitor the relevant aspects of the plan. Adaptation can be enhanced by including failure diagnostics capabilities beyond monitoring. In this way, the system not only knows that a discrepancy has occurred, but it could also reason *why* it has occurred, and enhance the decision process for the next steps of the execution with that information.

V. Discussion

This work-in-progress paper proposed a framework for autonomous robots where the basic functions of perception, planning and execution are enhanced by an ontology-based knowledge representation and a reasoning core that allows the framework to be wholly aware, i.e. aware of the situation (e.g. of the objects in the environment, their features, their relative locations), aware of the domain (e.g. of the predicates describing the state, of the actions and their preconditions and effects), and aware of the execution (e.g. of the alternative execution structures, of execution recovery strategies).

Such an awareness capability shall allow the framework to achieve a dynamic, robust and reliable adaptive behavior able to: a) automatically set the planning problem by reasoning on the initial and goal situations and on the domain, b) automatically configure the execution by reasoning on the execution structures and on the domain actions, and, c) automatically tune the monitoring procedures by reasoning on the task execution structure and on the current and desired situations.

The BE-AWARE framework is currently being implemented. The ontologies are being built using Protégé (https://protege.stanford.edu/), and they are integrated with the robotic systems in ROS through Owlready2 [22], using SPARQL to query and update the KB.

The challenges in the implementation of the proposed framework are:

- Flexibility and generality: the implementation needs to be transferable to different tasks, robots, and environments.
- Explainability: the reasoning behind the decisions taken by the framework must be traceable, explainable and interpretable by humans.
- Modularity: the different modules should be integrated with one another, but it is desirable to have modularity so that each particular one can be made available to the broader research community and integrated in other works.

References

[1] T. Haidegger, M. Barreto, P. Gonçalves, M. K. Habib, S. K. V. Ragavan, H. Li, A. Vaccarella, R. Perrone, and E. Prestes, "Applied ontologies and standards for service robots," *Robotics and Autonomous Systems*, vol. 61, pp. 1215–1223, 11 2013.

[2] B. M., F. D., and R. A., "Toward a framework for levels of robot autonomy in human-robot interaction," *Journal of Human-Robot Interaction*, vol. 3, p. 74, 7 2014. [Online]. Available: https://dl.acm.org/doi/10.5898/JHRI.3.2.Beer

[3] "IEEE recommended practice for assessing the impact of autonomous and intelligent systems on human well-being," *IEEE Std 7010-2020*, pp. 1–96, 2020.

[4] "Challenges of AI - Explainability — Feature from King's College London." [Online]. Available: https://www.kcl.ac.uk/challenges-of-ai-explainability

[5] P. Pirnay-Dummer, D. Ifenthaler, and N. M. Seel, *Knowledge Representation*. Boston, MA: Springer US, 2012, pp. 1689–1692. [Online]. Available: https://doi.org/10.1007/978-1-4419-1428-6_875

[6] A. Olivares-Alarcos, S. Foix, S. Borgo, and G. Alenyà, "OCRA – an ontology for collaborative robotics and adaptation," *Computers in Industry*, vol. 138, p. 103627, 6 2022.

[7] M. Diab, A. Akbari, M. U. Din, and J. Rosell, "PMK—A Knowledge Processing Framework for Autonomous Robotics Perception and Manipulation," *Sensors 2019, Vol. 19, Page 1166*, vol. 19, p. 1166, 3 2019.

[8] O. Ruiz-Celada, P. Verma, M. Diab, and J. Rosell, "Automating adaptive execution behaviors for robot manipulation," *IEEE Access*, vol. 10, pp. 123 489–123 497, 2022.

[9] M. Beetz, G. Kazhoyan, D. Vernon, and S. Member, "The CRAM Cognitive Architecture for Robot Manipulation in Everyday Activities," 4 2023. [Online]. Available: https://arxiv.org/abs/2304.14119v1

[10] A. Kattepur and B. Purushotaman, "RoboPlanner: a pragmatic task planning framework for autonomous robots," *Cognitive Computation and Systems*, vol. 2, pp. 12–22, 3 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1049/ccs.2019.0025

[11] M. Beetz, D. Bessler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, and G. Bartels, "KnowRob 2.0 - A 2nd Generation Knowledge Processing Framework for Cognition-Enabled Robotic Agents," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 512–519, 9 2018.

[12] T. Gruber, *Ontology*. New York, NY: Springer New York, 2016, pp. 1–3.

[13] "IEEE Standard for Autonomous Robotics (AuR) Ontology," *IEEE Std 1872.2-2021*, pp. 1–49, 2022.

[14] S. H. Joo, S. Manzoor, Y. G. Rocha, S. H. Bae, K. H. Lee, T. Y. Kuc, and M. Kim, "Autonomous Navigation Framework for Intelligent Robots Based on a Semantic Environment Modeling," *Applied Sciences 2020, Vol. 10, Page 3219*, vol. 10, p. 3219, 5 2020.

[15] Z. Kootbally, C. Schlenoff, C. Lawler, T. Kramer, and S. K. Gupta, "Towards robust assembly with knowledge representation for the planning domain definition language (PDDL)," *Robotics and Computer-Integrated Manufacturing*, vol. 33, pp. 42–55, 6 2015.

[16] M. Stenmark and J. Malec, "Knowledge-based instruction of manipulation tasks for industrial robotics," *Robotics and Computer-Integrated Manufacturing*, vol. 33, pp. 56–67, 2015.

[17] 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, and H. Li, "A review and comparison of ontology-based approaches to robot autonomy," *The Knowledge Engineering Review*, vol. 34, 2019.

[18] A. Munir, A. Aved, and E. Blasch, "Situational Awareness: Techniques, Challenges, and Prospects," *AI 2022, Vol. 3, Pages 55-77*, vol. 3, pp. 55–77, 1 2022. [Online]. Available: https://www.mdpi.com/2673-2688/3/1/5

[19] S. Bøgh, O. Nielsen, M. Pedersen, V. Krüger, and O. Madsen, "Does your robot have skills?" Aug. 2012.

[20] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI* . CRC Press, 7 2018.

[21] P. Hayes, "The frame problem and related problems in artificial intelligence," *Readings in Artificial Intelligence*, pp. 223–230, 1 1981.

[22] J.-B. Lamy, "Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," *Artificial Intelligence in Medicine*, vol. 80, pp. 11–28, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365717300271