ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA241021

An Empirical Study of Grounding PPDDL Plans for AI-Driven Robots in Social Environment

Gloria Beraldo[a,*], Angelo Oddi[a] and Riccardo Rasconi[a]

[a]Institute of Cognitive Sciences and Technologies, National Research Council of Italy (CNR-ISTC)
ORCID (Gloria Beraldo): https://orcid.org/0000-0001-8937-9739, ORCID (Angelo Oddi): https://orcid.org/0000-0003-4370-7156, ORCID (Riccardo Rasconi): https://orcid.org//0000-0003-2420-4713

Abstract. Autonomous robots are agents that interact with the environment and perform tasks using their own abilities (i.e., skills) without continuous human intervention. However, in real-life scenarios, intelligent robots also need to discover the effects of their actions and understand how to save them for future use. This task appears time-consuming and very challenging, especially in a social environment populated by people who typically modify their behaviors based on the context and can dynamically impact the robot's decision-making process. This paper aims to investigate the feasibility of autonomously creating an abstract representation of the domain knowledge from the data acquired during the robot's exploration, inferring causal-effect relations between the executed actions, and learning context-aware symbols that describe the environment states at high level, ultimately producing a PDDL-based description of the domain. With this purpose, a new framework that relies on ROS, the standard de-facto in robotics, and ROSPlan has been developed to facilitate the transfer into several robotic platforms. Preliminary results suggest the possibility of describing the robot's experience per option via context-based symbols that are consistently learned by the system from a few data samples.

1 Introduction

A very important challenge that artificial agents must face is how to achieve symbolic reasoning from the low-level, noisy, and multidimensional data obtained from sensors. Having the ability to make high-level decision processes, while perceiving the environment (as well as to act on it) through the low level, is absolutely desirable [8].

A first level of abstraction from low-level control details can be achieved through techniques such as Hierarchical Reinforcement Learning [2], which enable the agent to learn and plan using sets of higher-level skills appropriately concatenated to achieve behaviors for specific tasks towards the attainment of properly identified goals [18]. One example of Hierarchical Reinforcement Learning framework is the *option framework* [20], in which the agent's actions are modeled in terms of *options*. Very briefly, an option o is characterized by three elements: an Initiation Set I_o which defines the states where the option can be initiated, a policy π_o which determines the action selection depending on the current state, and a Termination Condition β_o which defines the conditions under which the option terminates.

Despite this initial type of abstraction significantly enhances the reasoning capabilities of the artificial agent, the task of planning remains inherently difficult due to the fact that the state space remains continuous and highly multidimensional, making evident that a further step of abstraction is desirable. In [10] an algorithm has been proposed which, starting from a representation based on options, performs a further abstraction towards a fully symbolic representation of both the options and the representative states of the environment in terms of Planning Domain Definition Language (PDDL) [6] and its Probabilistic version PPDDL [23], a well-known language used in automated planning.

The advantage gained from obtaining a symbolic representation of the domain following this abstraction process is twofold. On the one hand, the explicitation of the causal relationships existing among all the skills (options) executable by the agent, in terms of preconditions and effects, which would have remained implicit (thus unexploitable) if expressed in the option framework; on the other hand, it offers the possibility to express both operators and goal states in a language immediately understandable to already available automated planners. As a net result, the synthesis of a symbolic representation of the domain endows the agent with the capability to: (i) express goals in high-level and abstract terms, and (ii) reason about the achievement of such goals, producing plans whose complexity exceeds the complexity achievable through the concatenation of the high-level skills in the option framework.

In this work, we are particularly interested in analyzing the potential benefits of information abstraction in the context of human-robot interaction. In particular, our analysis focuses on the advantages of autonomously building an abstract representation of the environment in which the workspace is shared between both robotic agents and humans, from the robotic agent's perspective. The main question this work addresses is the following: is it possible to autonomously acquire from direct experience a symbolic representation that embraces the human presence?

An Example Scenario: *imagine that a service robot acting as a worker has to carry objects between different locations in an office environment (see Figure 1). The environment is populated by people that can interfere with the robot's behaviour, e.g., they might get into the robot's way during its deambulation activities. The robot can learn how to interact with them in order to reach its final goal. Indeed, within this social scenario the robot can receive a high-level goal from a human (e.g., move one object from a place to another in the office) and during the execution of the related commands, it*

* Corresponding Author. Email: name.surname@cnr.it

could experience new events that may be useful to optimise its future behaviour. For example, a failure could happen, because the operator does not know that the scenario situation has changed (e.g., there are new prohibited areas where the robot cannot enter, as the door pathway is obstructed). In general, the coexistence with humans can generate many chances of acquiring new knowledge from the experience and a social scenario can work as a training environment for the robot, resembling in some sense the framework of an intuitive robot programming [22].

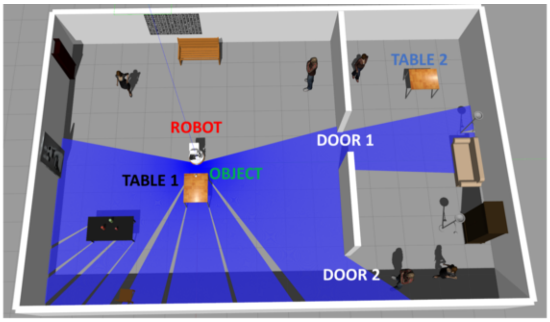

Figure 1. The illustrative social environment used in this work, featuring a robotic agent that (i) navigates within an office setting complete with common furniture items such as lamps, chairs, sofas, and tables, and (ii) interacts with people. The robotic agent engages in tasks such as handling objects by picking them up, navigating through the existing doorways, and placing them on various tables.

The contribution of the paper is:

- a novel ROS-based framework for developing and assessing cognitive architectures for controlling robotic agents, that allows to integrate option learning and symbolic reasoning in multiple domains;
- the definition of a Gazebo environment that features a Tiago robot equipped with a 7-DOF arm and a two-fingers gripper, which can move around, pick/place objects and interact with people;
- an empirical investigation in a social environment of the feasibility of extending the PPDDL-based domain representation through an autonomous information abstraction process [10].

The remaining paper is organized as follows. Section 2 describes some related works, whereas Section 3 introduces the reader to the necessary background. Section 4 describes the used methodology, whereas the empirical validation of the proposed framework for option learning and symbolic abstraction is described in Section 5. Finally, Section 6 and 7, respectively, close the paper with a final discussion and some conclusions.

2 Related Work

There are several works that investigate how to learn actions models that can be used for autonomous robots [1, 3, 5, 7, 9, 13]. However, differently to this work, they do not face the additional challenge of learning from realistic and uncertain robot's perception, which is what typically happens in real world scenarios that evolve over time and may hinder the correct execution of the robot's actions.

More similar to our idea are, for example, approaches based on reinforcement learning such as [19, 14, 15], where the final robot's goal is provided in input a priori, and the robot learns how to reach it by dynamically interacting with the environment. The final result is a policy, describing the robot's experience, that can be used in the next interaction. Typically these approaches suffer from two limitations: (a) the possible lack of interpretability for a human due to the nature of the robot's knowledge representation (i.e., policy); (b) the need of an extensive exploration phase for acquiring the necessary data for the learning. Herein, on the one hand, this work aims at describing the acquiring robot's knowledge experienced in terms of causal preconditions and effects of the options. On the other hand, we are facing an additional challenge consisting of the abstraction process based on a relatively small number of samples.

Furthermore, it is important to highlight that in this context, our focus lies on understanding the causal relationships among the events encountered by the robot. This differs from other studies which primarily seek to gather useful features for planning purposes. Here, our objective is to gather strategic insights to augment the robot's initial knowledge base. In particular, the DISCOVER-PLAN-ACT paradigm introduced in [16] inspired our work. However, here we focus on spatial HRI; the human presence can impact on the robot's actions and hence on its next choices. The acquisition of new information is achieved via human-robot interactions and robot's experiences within an uncertain and noisy social environment. These interactions are automatically generated by the proposed framework and are represented symbolically. This "social" mechanism adds complexity and is not explored in [16] where the environmental setup is not affected by any other agent but the robot.

With the exclusion of context and methodology, in a similar fashion, the works [21, 12] propose approaches to learn the predicates corresponding to action effects after their executions. To facilitate the transfer into a real robot as well as to other robotic platforms, our framework is completely integrated into the Robot Operating System (ROS), the standard de facto in robotics, and designed to be as realistic as possible, avoiding the utilization of oracles or similiar, that provide the robot with contextual information. The robot autonomously senses, plans and acts driven by a high-goal, provided in input by a human (e.g., move an object to another place) and represented by a initial PDDL problem that actually does not describe all the possible situations that may arise during the task (e.g., transportation) and allows the robot to experience new situations, generally unforeseen by the people that provide the high-goal, which could lead to failures. With respect to the current state-of-the-art approaches like [16], the complexity increases to a level that can no longer be fully represented by the PDDL language. Hence, the need to resort to the probabilistic version of PDDL (i.e., PPDDL) [23] to better capture the aspects related to uncertainty.

3 Background

For reasons of self-containment, in this section we provide a succinct description of the abstraction procedure used to synthesize the symbolic representation of the agent's domain in terms of a propositional model. The abstraction procedure is mainly composed of two steps. In the first step, low-level data are acquired from the execution of the agent's skills that will serve as input for the second step, in which the actual abstraction process is executed. The details of the overall abstraction procedure are beyond the paper's scope; the interested reader may refer to [10] for further details.

3.1 The Low-level Data Acquisition Procedure

In this step, the agent is supposed to be capable of executing a fixed number of skills (henceforth referred to as *options* [20]). The agent starts executing such options in the environment, i.e., in a randomic fashion or according to a specific policy. For every execution of a skill a set of low-level data acquired through the agent's sensors are properly saved; these data constitute the input of the abstraction procedure. In particular, two types of data are collected for each option o_i: (i) the *Initiation Data*, collecting the low-level state s_i from which o_i is run (it is here assumed that the agent is able to determine whether the option can be executed from the current state s_i), and (ii) the *Transition Data*, collecting the tuple (s_i, o_i, r_i, s'_i) where s'_i is the termination state reached after o_i's execution, and the agent is able to determine whether the option has successfully terminated or not, depending on the respectively positive/negative value of r_i.

3.2 The Abstraction Procedure

The PPDDL encoding construction procedure uses the low-level data obtained in the previous phase; the four steps that constitute the procedure are informally sketched in the following:

1. *Subgoal option partition*: each option is partitioned in the set of the corresponding *subgoal options* (see [17]), that is, the options that are characterized by the same distribution of termination states regardless of the distribution of the initial states (i.e., the low-level states the option starts from). This step essentially solves a clustering problem (e.g., using DBSCAN).
2. *Learning of precondition classifiers*: in this step, the agent learns a precondition classifier Cl_o^{pre} for each subgoal option o. Each subgoal option is characterized by a set of termination states, corresponding to a set of initial states; the precondition classifier is trained using the termination states as positive examples, while all the other states are used as negative examples. The classification can be performed using a SVM fitted only on the relevant low-level variables of the states (those that change, i.e., features).
3. *Computation of effect distributions*: in this step, the effect distribution of each subgoal option is computed using a Density Estimator (e.g., Gaussian kernel). The effect distribution of each subgoal option is learned by the agent based on the elements that are modified by the option's execution. This step is of particular importance as the returned effect distributions define the whole propositional vocabulary P that will be translated later into PPDDL terms.
4. *Computation of precondition distributions*: from the previous step 3 we see that the effects are defined in terms of the propositional vocabulary P, while the preconditions are expressed in terms of classifiers (see step 2). It is therefore necessary to express the preconditions in the same terms as the effects. In this step, an operation is performed in which, for each subgoal option o, the agent considers all the possible combinations of the effect state distributions of o, and extracts the samples of the conjunctions of each combination. Given subgoal option o's precondition classifier Cl_o^{pre}, if a sample is positively classified by Cl_o^{pre}, then the combination of the effect state distribution associated with the sample is used to represent o's precondition.

At the end of this procedure both preconditions and effects for each option o_i are defined as proposition in P, which allows the formulation of a sound and complete PPDDL representation of the agent's domain. In the next sections, we describe how the capability of autonomously abstracting a symbolic representation of the environment can be exploited to facilitate the "cohabitation" of both human and robotic agents in a shared environment.

4 Methodology

4.1 Framework Overview

Our system relies on a three-tier architecture composed of three layers plus the abstraction procedure (see Figure 2a), described below:

- the *Decision layer* is in charge of planning the robot's actions based on the goal chosen by a human operator and the robot's skills. The initial knowledge is represented by a high-level PDDL domain and problem that drive the robot's decision making process. The initial PDDL-based domain is supposed not to cover all the situations the robot may experiment during the real execution, possibly due to the incomplete domain knowledge on behalf of the human modeller. For instance, in reference to the tested scenario, imagine that the passage through the door is not possible in certain times due to the presence of an occlusion. Such event may not be considered and modelled in the initial PDDL domain description;
- the *Executive layer* is responsible for receiving the chosen robot's action from the *decision layer* and translating it into robot's behaviors that activate the required low-level actuators and sensors. For instance, imagine that the robot's is asked to pick an object, the robot's needs to perceive the area of interest with the possibility of moving the robot's camera, detects the objects and then starts the picking routines planned by a motion planner based on the collected information;
- the *Functional layer* is associated with the low-level sensors for context perception, as well as to the robot's actuators;
- the *Abstraction Procedure*, described in Section 3.2, classifies the contextual environmental states acquired during the robot's execution of the chosen actions based on the initial high-level PDDL and creates an enriched PPDDL with the corresponding symbols synthesized based on the encountered successes and failures.

As mentioned before, the proposed framework is integrated within the ROS ecosystem. Therefore, we chose to exploit ROSPlan[1] [4] for the *decision layer* given the affinity and the consensus it has received from the planning & robotics community. The possible robotic skills, presented in detail in Section 4.2, are implemented via ROS actions that can return a positive or negative feedback based on their execution. In the current implementation, for simplicity, in case of failure, re-planning is not applied. In this way, the output plan is deterministic independently by the actions' execution. Only the feedback (i.e., success vs. failure) and the contextual environmental state — data that are necessary for the *functional layer* — are acquired at the execution's start and end of every option.

The *execution layer* implements context-based routines that are called by the *deliberative layer* (i.e., ROSPlan) using the Request/Reply communication modality via ROS-services. Differently from the *deliberative layer*, at this stage, the execution can vary over the runs due to the real-time robot's perception and actuators. For instance, in the experimented scenario, the robot that is required to reach a specific position can arrive in different ways in proximity of the target. The same can happen for the manipulation of objects. Based on the motion planner output, the robot's joints can change their positions run per run even if the outcome (i.e., success vs. failure) is the same. In this work, the low-level executor runs on the *Gazebo* simulation completely integrated into ROS. However, it is worth noting that the executed robot's motions and perception do not use any basic simulated motion and/or joystick input, but are rather handled autonomously by the robot based on the current perceived state, as

[1] https://kcl-planning.github.io/ROSPlan/

in a real world scenario. The reader may refer to Section 4.2 for the detailed implementation of the robotics skills.

Finally, the *functional layer* extends the implementation of the abstraction procedure proposed by [10] from a collection of environmental states that are provided in input in the form of *Initiation data* (i.e., the information describing the initial state of each option and the possible option to execute from it) and the *Transition data* (i.e., the information describing the changes in the variables describing the environmental state between the start and the end of the option, the outcome (i.e., success vs. failure), the completion).

4.2 Robot & Robotic Skills

In this work, we are focusing on TIAGo[2] by PAL robotics, a service robot that is designed to navigate, manipulate objects and socially interact with people. However, the same framework can run to other comparable robotic platforms (in terms of options) since it is integrated into ROS. TIAGo has a differential mobile base and it is endowed with a 2D lidar and a RGB-D camera (placed on its head) for the perception. For manipulation, TIAGo is characterised by a 7 DoF-arm and a two-fingers gripper. In this work, TIAGo is able to autonomously implement the following skills (see Figure 2):

- *GOTO* allows the robot to move from the current position to the destination provided in input as a parameter. The robot's trajectory is computed online and optimized by using the perception from both the lidar and the camera based on the *Dynamic Window Approach* (DWA) as a global motion planner and *Timed Elastic Band* (TEB) as local motion planner inside the ROS *navigation stack*[3]. The robot also exploits a map of the social environment that has been acquired in a preliminary phase based on *GMapping* as SLAM algorithm. If a dynamical obstacle is perceived in the robot's computed path, the robot tries to re-compute a new trajectory connecting the current position to the final destination.
- *PICK* consists of different sub-phases. First, the robot observes the working space in front of it, by lowering its head down (i.e., also the camera) and turning it left and right. The detection of the object is based on the RGB-D camera stream and the *arUco markers*[4]. In other words, the object has a marker on its top. When the maker is detected by the robot, its position is computed in the world reference frame. Then, the motion planner based on *MoveIt*[5] is activated and computes the positions for the joints in order to reach the object, the different grasp positions and finally executes the motion. The gripper fingers are hence opened and closed and, at the end, the object is lifted up based on a pre-registered motion.
- *DOOR PATHWAY* allows the robot to move from the current position to one of the doors that is provided in input. The robot's navigation is handled as described in the *GOTO*. However, based on its perception, if the target door is occupied due to the presence of a perceived obstacle (e.g., a person), it tries to verbally interact before re-planning the trajectory.
- *CARRY* allows the robot to move from the current position to a destination that is provided in input as a parameter. The robot's navigation is handled as described in the *GOTO*. Differently, in the *CARRY*, it is expected that the gripper is grasping an object.
- *PLACE* frees the robot's gripper. Specifically, the robot opens the gripper's fingers based on a pre-recorded motion.

[2] https://pal-robotics.com/robots/tiago/
[3] https://wiki.ros.org/navigation
[4] https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
[5] https://moveit.ros.org/

Figure 2(b) shows the evolution of the Tiago's perception and action per each option. It is worth highlighting that, although the framework runs in the Gazebo simulation, both the perception and the execution tasks are prone to non-deterministic outcomes, as in a real world scenario. For instance, notice the slight misalignment between the physical object and the detected 3D bounding box around it during the *PICK* behaviour. This is a challenging aspect that we are facing in this work and increases the complexity both in the successful execution and in the abstraction process.

4.3 Data Collection For Abstraction

Data collection for the abstraction process is autonomously performed following the plan found by the *deliberative layer* implemented via ROSPlan. Specifically, in our scenario in Figure 1, the high-level goal for the robot in the PDDL domain is to pick the object on the table 1 in the first room and place it on the table 2 by going through one of the doors chosen by the human operator. The PDDL knowledge domain considers only the preconditions for the PICK and the PLACE operators in terms of robot's proximity to the target positions and the presence of the object first on the table and then on the robot: However, different scenarios have been explored in this work during the target collection that might lead to a failure in the plan execution because they are not described in the initial PDDL. In particular, we have been focused on the passage through the door that is considered as a challenging navigation task in robotics. For instance, the failure can be caused by the presence of an entity that obstructs the targeted door. Moreover, due to both the noisy perception and the uncertainty of the robot's behavior (e.g., the robot may reach the same positions in several ways) failures can arise also in the best case (i.e., the target door is free). Thus, the data collection includes both positive and negative examples (i.e., based on the high-level goal reaching) that are realistic and not artificially generated.

Listing 1. PICK and PLACE operators in the initial PDDL domain.

```
(:action pick
        :parameters (?g – gripper ?wp – waypoint ?ob – object)
        :precondition (and
                (robot_at ?wp)
                (object_on_table ?wp ?ob)
                (free ?g))
        :effect (and
                (object_on_robot ?g ?ob)
                (not (object_on_table ?wp ?ob))
                (not (free ?g)))

)
(:action place
        :parameters (?g – gripper ?wp – waypoint ?d – waypoint
            ↪ ?ob – object)
        :precondition (and
                (robot_at ?wp)
                (object_on_robot ?g ?ob)
                (not (object_on_table ?wp ?ob))
                (not (free ?g)))
        :effect (and
                (object_on_table ?wp ?ob)
                (free ?g)
                (not (object_on_robot ?g ?ob)))
)
```

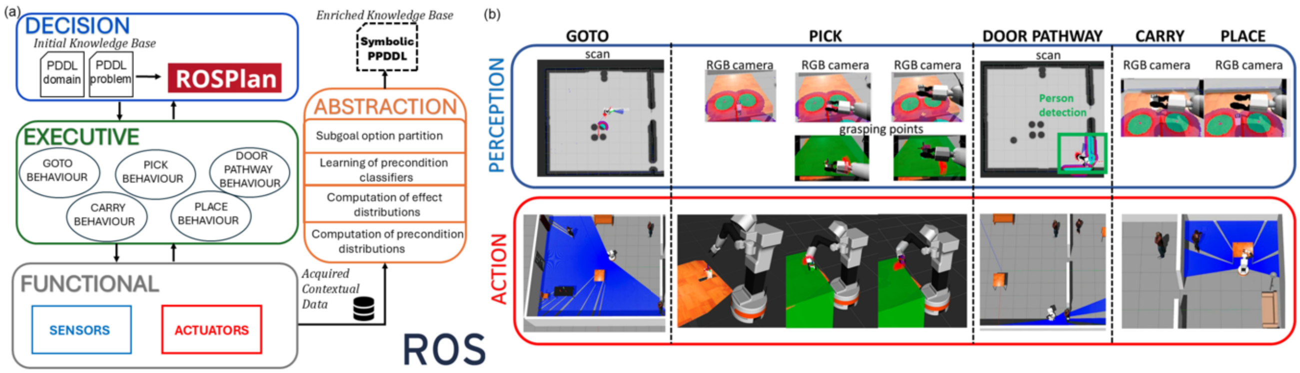

Figure 2. (a) A sketch of the system's architecture. (b) Illustrative representation showing the skills that the robot can perform in autonomy and the related perception. Note that the perception driven the robot's execution of the options is realistic and affected by noise as much as possible as in a real world.

Moreover, given such an uncertainty in the onboard robot's sensors, we assume that in a real scenario, the social office is domotic and further sensorised with additional environmental sensors that can detect the door occupancy state at any moment, and in particular the presence of a human at the door.

The low-level variables that we consider in the contextual environmental state collected at the beginning and the end of the execution of each option are the following:

- the robot positions (x_r, y_r, z_r);
- the gripper positions (x_g, y_g, z_g);
- the object position (x_o, y_o, z_o);
- a binary value that is equal to 1 if the gripper holds the object, 0 otherwise;
- a binary value that is equal to 1 if the target door is occupied; 0 otherwise;
- a binary value that is equal to 1 if the target door is occupied by a person; 0 otherwise.

Such information, together with the actions execution outcomes (success vs. failure) are stored in two .csv files representing respectively the *Initiation data* and the *Transition data*.

5 Empirical Evaluation

The main purpose of our empirical evaluation is to verify the feasibility of dynamically acquiring new information by the robot through the experience that can be abstractly represented via context-aware symbols. In particular, the tested social environment in Figure 1 evolves over time due to the presence of entities at the targeted door passage that dynamically impact the robot's decision-making process. We focus on two kinds of occlusions that are shown in Figure 3:

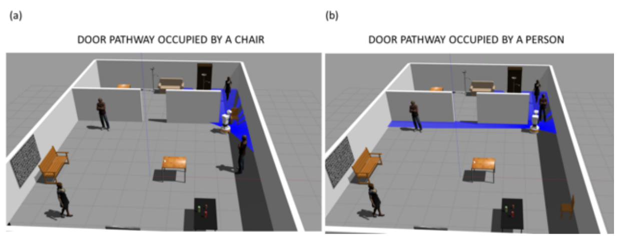

Figure 3. Illustrative figure representing the possible situations of door pathway occlusion by respectively a chair or a person.

(a) a static object like a chair, typically confused by the robot as a person from a perception point of view, that remains in the same position; (b) a person that moves after the interaction with the robot to let it pass through the door. In both cases, at the *execution layer*, the robot autonomously executes the same behaviors described in Section 4.2. When, it is autonomously inferred (based on perception) that it is impossible to pass through the targeted door, e.g., as it is obstructed by a chair, the motion planner re-actively tries to re-compute a new trajectory towards the destination (typically, the robot goes through the other free door to enter the second room). However, also in such a situation, at the end, the robot fails in reaching the expected final destination since it is still occupied by the chair. In other words, the modalities of door occlusion tested lead a two different outcomes: (a) always failure when there is the chair; (b) success when the person is there and no other failures occur (e.g., drop of the object). Note that such situations are not described in terms of preconditions in the initial PDDL domain (see Section 4.3).

Four testing conditions have been considered: (a) free door 1 as target; (b) free door 2 as target; (c) door 2 as a target occupied by a chair; (d) door 2 as a target occupied by the human. Overall, only 25 runs in total (i.e., episodes) for each experimented conditions (i.e., 100 in total) have been recorded and provided in input to the *functional layer* for the abstraction process. The experiments were run on a Desktop computer (12th Gen Intel® Core™ i9-12900F × 24, NVIDIA GeForce RTX 4070, 32 GB RAM), based on Ubuntu 20.04LTS and ROS-Noetic, CUDA Version 12.3).

5.1 Results

The *functional layer* produced a PPDDL representation (both domain and problem) from the contextual environmental states collected by the robot during the 100 runs. The resulted PPDDL domain is composed of 167 symbols and 2562 operators. This result might suggest that the robot's knowledge base is significantly enriched with respect to the initial one, since the PDDL provided as input to ROSPlan relies on only 4 symbols and 5 operators.

To better analyse the output, given the space constraint, we focus on the DOOR PATHWAY option that is the most interesting in our social scenario given the presence of two kinds of occlusion that impact on the successful reaching of the robot's high-level goal (i.e., place the object on table 2). In the produced PPDDL domain, we can recognize at least the presence of four different *subgoal option partitions* (i.e., represented as operators) that result from the clustering operation behind the abstraction process (see Section 3.2) and describe each of the experimented situations in terms of initial and effect con-

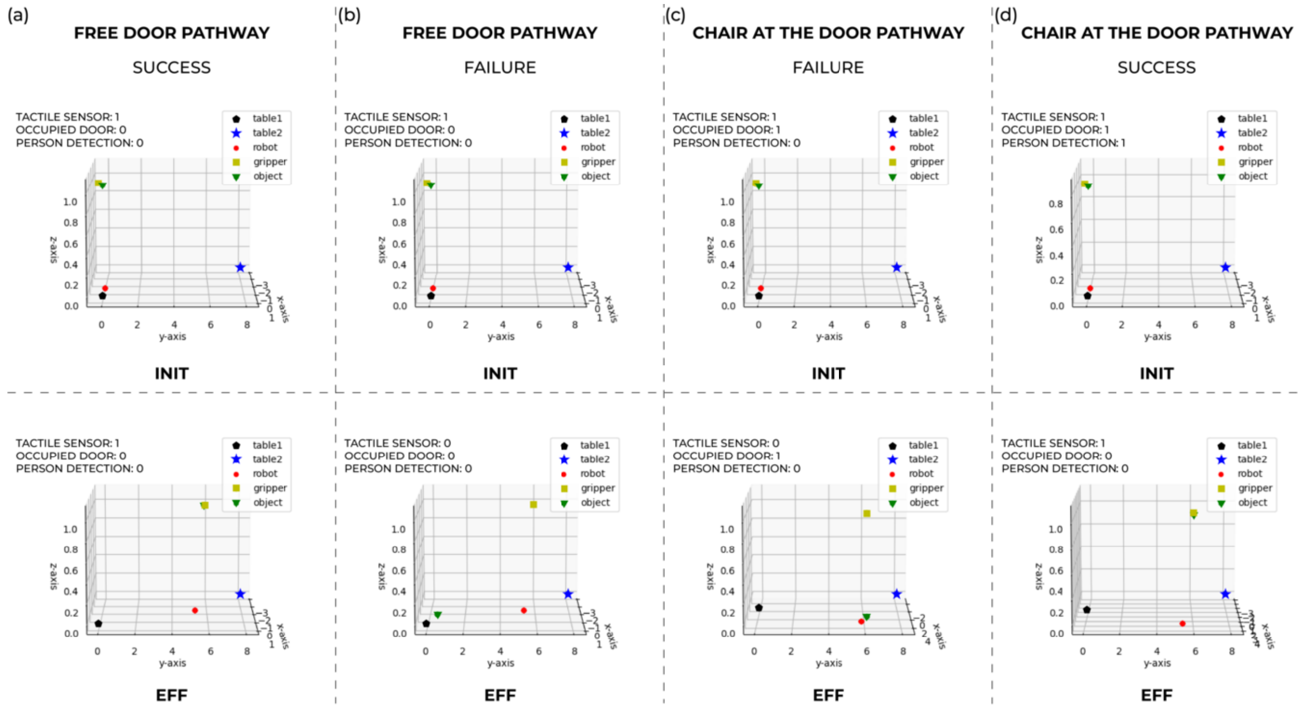

Figure 4. A 3D graphical representation of the corresponding contextual environmental states (see Section 4.3) associated with the examined *partitions* (top: initial state; bottom: effect state) in the following cases: (a) the target door pathway is free (success); (b) the target door pathway is free (failure); (c) the door pathway is occupied by the chair (failure); (d) the door pathway is occupied by the person (success).

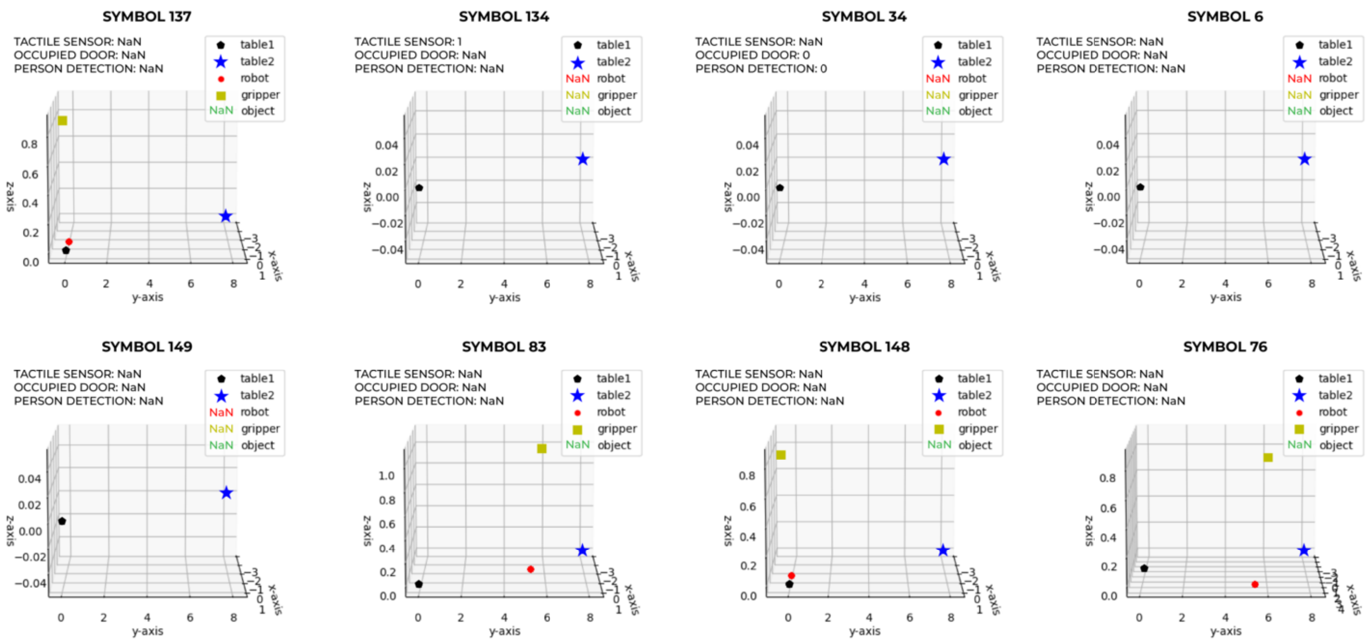

Figure 5. A 3D graphical representation of some of the symbols achieved from the abstraction process in the examined operators of the symbolic PPDDL.

textual states. A 3D graphical representation of the corresponding contextual environmental states (i.e., robot's position, gripper's position, object's position, tactile sensor output, occupied door output, person detection output) as described in Section 4.3 associated with the mentioned *partitions* is shown in Figure 4 for the following cases: (a) the target door pathway is free (success outcome); (b) the target door pathway is free (failure outcome since the robot drops the object); (c) the door pathway is occupied by the chair (failure outcome, notice that the robot also drops the object); (d) the door pathway is occupied by the person (success outcome since the person moves into an another position to let the robot pass through the door). As you can notice by observing the changes in the low-level variables, the results are perfectly consistent with the expected situations experimented by the robot. Furthermore, Figure 4 also highlights that, for instance, the robot's base and robot's gripper position are not always the same despite the same outcome due to the realistic robot's behaviours executed at the *execution layer*.

Now, let's focus on the meaning of the symbols derived by the abstraction process for instance in the two following operators associated respectively with the free door reported in Listing 2 vs. the presence of the person reported in Listing 3.

Listing 2. Extract of one of the grounded operators associated with the passage through a free door in the symbolic PPDDL.

```
(:action door_passway-partition-0-13
    :parameters ()
    :precondition (and (notfailed) (symbol_137) (
        ↪ symbol_134) (symbol_34))
    :effect (probabilistic
                        0.404 (and (symbol_6)
                        ↪ (
                        ↪ symbol_149)
                        ↪ (symbol_83)
                        ↪ (not....
```

Listing 3. Extract of one of the grounded operators associated with the passage through a door occluded by a person in the symbolic PPDDL.

```
(:action door_passway-partition-10-925
    :parameters ()
    :precondition (and (notfailed) (symbol_148) (
        ↪ symbol_145) (symbol_30))
    :effect (probabilistic
                        0.819 (not (notfailed))
                        0.181 (and (
                        ↪ symbol_34) (
                        ↪ symbol_2) (
                        ↪ symbol_75) (
                        ↪ symbol_76) (
                        ↪ not ...
```

To facilitate the human understanding, we exploit the same 3D graphical representation used above for the involved symbols in the examined *partitions*. We represent only the low-variables in the environmental state that change based on the symbol they represent. Figure 5 shows some of the grounded symbols reported in the preconditions and the effects of the analysed operators. *Symbol_145*, *Symbol_30*, *Symbol_2*, *Symbol_75* were omitted because they have the same semantic meaning of *Symbol_6*. The same is true for *Symbol_6* and *Symbol_149* that are reported to show with an example the limitation of the current abstraction process in terms of generation of spurious symbols. Except for this aspect, the symbols appear in line with the associated option.

6 Discussion

Human-robot interaction scenarios pose significant research challenges, often pushing state-of-the-art AI systems to their limits. This paper adopts the perspective that each social scenario constitutes an *open* environment, demanding robotic agents to continuously acquire new skills and uncover new effects of existing capabilities. Indeed, the same option, when performed in different contexts or seemingly similar contexts with subtle differences, can yield different outcomes. For instance, in our proposed office domain, when a gate is occluded, the robot learns that people are always willing to free the door passage, whereas the presence of one object (e.g., a chair) leads to the failure of the corresponding option.

One of the main challenges to address in robotics is achieving *long-term autonomy* (LTA) [11], it involves the ability of a robot to sustain task execution over extended periods in dynamic and uncertain environments, while autonomously accumulating environmental knowledge. We view the utilization of social training, as presented in this paper, as a potential catalyst for achieving long-term autonomy. In essence, the social environment should serve as the primary source of knowledge for the robot. These concepts resemble the framework of *intuitive robot programming* (IRP), which aims to simplify robot programming and interaction through intuitive interfaces and methods. Hence, in a social setting, individuals can act both as *trainers*, imparting essential environmental skills to the robot, and as occasional users, when a robot operates within real social contexts. Both can enrich the robot's knowledge about the working scenario.

A careful analysis of the empirical evaluation proposed in Section 5 reveals the presence of additional challenges to solve. Firstly, we observe that the real world, where a robot operates is intrinsically uncertain. For example, a grasped object may fall or a robot can reach a goal position in several ways. As a consequence, the abstraction process described in Section 3 could generate a large set of different symbols and operators, and this fact is mainly due to the problem of abstracting a dataset of continuous state values - i.e., both the initial and the transition dataset - into a discrete set of symbols and operators. In general, this process requires a no trivial tuning of the used machine learning algorithms [10]. In addition, we observe that the knowledge acquired by the robot must be externally accessible to the human trainers/users, enabling them to understand the semantics of generated symbols and set new goals for the robot. In general, the automatic generation of this kind of information (e.g, via a graphic representation, see for example Figure 4 or Figure 5) constitutes a further challenge to solve. To improve the interpretability of the obtained symbolic PPDDL, the authors plan to investigate the integration of more human-like representations like semantic-based technologies (e.g. ontologies) or learning methods, therefore more closely mapping the current symbols to the human terminology.

7 Conclusion

In this paper, we have examined the potential advantages of information abstraction within human-robot interaction scenarios. Specifically, our investigation centers on the benefits of autonomously constructing an abstract model of a shared workspace environment from the perspective of the robotic agent. To facilitate this exploration, we developed a novel Gazebo environment featuring a TIAGo robot equipped with a 7-DoF arm and a two-finger gripper, which can move around the environment, pick/place objects and interact with people. Our environment is assessed within a novel ROS-based framework designed for the creation and evaluation of cognitive architectures for robotic agent control. This framework is capable of incorporating option learning and symbolic reasoning across various domains. Our empirical findings regarding the feasibility of grounding a PPDDL domain via abstraction demonstrate the system's ability to construct a PPDDL representation that captures domain uncertainty, consistently representing the newly acquired knowledge, in particular the operators' preconditions, during the training phase.

Acknowledgements

This work is supported by PNRR MUR project PE0000013-FAIR.

References

[1] D. Aineto, S. Jiménez Celorrio, and E. Onaindia. Learning action models with minimal observability. *Artificial Intelligence*, 275:104–137, 2019. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2019.05.003. URL https://www.sciencedirect.com/science/article/pii/S0004370218304259.

[2] A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13:41–77, 2003. URL https://api.semanticscholar.org/CorpusID:386824.

[3] B. Bonet and H. Geffner. Learning first-order symbolic representations for planning from the structure of the state space. In *ECAI 2020*, pages 2322–2329. IOS Press, 2020.

[4] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos, and M. Carreras. ROSPlan: Planning in the Robot Operating System. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 25, pages 333–341, Apr. 2015. doi: 10.1609/icaps.v25i1.13699. URL https://ojs.aaai.org/index.php/ICAPS/article/view/13699.

[5] S. N. Cresswell, T. L. McCluskey, and M. M. West. Acquiring planning domain models using LOCM. *The Knowledge Engineering Review*, 28 (2):195–213, 2013. doi: 10.1017/S0269888912000422.

[6] M. Ghallab, A. Howe, C. Knoblock, D. Mcdermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL—The Planning Domain Definition Language, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.212.

[7] M. Grand, H. Fiorino, and D. Pellier. Amlsi: A novel accurate action model learning algorithm. *arXiv preprint arXiv:2011.13277*, 2020.

[8] M. K. Ho, D. Abel, T. L. Griffiths, and M. L. Littman. The value of abstraction. *Current Opinion in Behavioral Sciences*, 29:111–116, 2019. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2019.05.001. URL https://www.sciencedirect.com/science/article/pii/S2352154619300026. Artificial Intelligence.

[9] B. Juba, H. S. Le, and R. Stern. Safe Learning of Lifted Action Models. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, pages 379–389, 11 2021. doi: 10.24963/kr.2021/36. URL https://doi.org/10.24963/kr.2021/36.

[10] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez. From skills to symbols: learning symbolic representations for abstract high-level planning. *J. Artif. Int. Res.*, 61(1):215–289, jan 2018. ISSN 1076-9757.

[11] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajník. Artificial intelligence for long-term robot autonomy: A survey. *IEEE Robotics and Automation Letters*, 3(4):4023–4030, 2018. doi: 10.1109/LRA.2018.2860628.

[12] L. Lamanna, M. Faridghasemnia, A. Gerevini, A. Saetti, A. Saffiotti, L. Serafini, and P. Traverso. Learning to act for perceiving in partially unknown environments. In E. Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 5485–5493. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/609. URL https://doi.org/10.24963/ijcai.2023/609. Main Track.

[13] T. L. McCluskey, S. N. Cresswell, N. E. Richardson, and M. M. West. Automated acquisition of action knowledge. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 93–100, 2009. doi: 10.5220/0001662500930100.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, 2015. doi: https://doi.org/10.1038/nature14236.

[15] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1928–1937. JMLR.org, 2016.

[16] A. Oddi, R. Rasconi, V. G. Santucci, G. Sartor, E. Cartoni, F. Mannella, and G. Baldassarre. Integrating open-ended learning in the sense-plan-act robot control paradigm. In *ECAI 2020*, pages 2417–2424. IOS Press, 2020.

[17] D. Precup and R. S. Sutton. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 2000. AAI9978540.

[18] D. Silver and K. Ciosek. Compositional planning using optimal option models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1267–1274, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[19] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

[20] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[21] E. Ugur and J. Piater. Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2627–2633. IEEE, 2015. doi: 10.1109/ICRA.2015.7139553.

[22] J. Wassermann, A. Vick, and J. Krüger. Intuitive robot programming through environment perception, augmented reality simulation and automated program verification. *Procedia CIRP*, 76:161–166, 2018. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2018.01.036. URL https://www.sciencedirect.com/science/article/pii/S2212827118300507. 7th CIRP Conference on Assembly Technologies and Systems (CATS 2018).

[23] H. L. S. Younes, M. L. Littman, D. Weissman, and J. Asmuth. The first probabilistic track of the international planning competition. *J. Artif. Intell. Res.*, 24:851–887, 2005. doi: https://doi.org/10.1613/jair.1880.