

Study of the Anchoring Problem in Generalist Robots based on ROSPlan

Diego ESCUDERO-RODRIGO ^{a,1}, René ALQUEZAR ^a,
^a *Institut de Robotica i Informatica Industrial, CSIC-UPC, Spain*

Abstract. In real life environments where robots must deal with complex situations and humans, *generalist robots* that adapt to novel situations are needed. They are composed by two sub-systems: perception/actuation and knowledge representation, and they need that symbols in the high-level area are coupled to objects and actions of the low-level area. This is the so-called *Anchoring Problem*. In this paper we present the system we are using to study this problem. It is based on ROSPlan, a framework that provides a generic method for task planning in a ROS system. The high-level area is composed by a planner that uses PDDL files and a knowledge representation system, while the low-level area is defined as a set of robot services exported using ROS actions, services and topics. We plan to contribute to this problem by applying human-robot interaction and learning techniques, and our main objectives are: (1) link an existing symbol with a learned action by interaction, and (2) automated code generation of ad-hoc ROS nodes that connect symbols to specific perceptions/actions.

Keywords. anchoring problem, AI planning, learning, PDDL, action grounding, human-robot interaction, ROS, generalist robots

1. Introduction

In real life environments where robots must deal with complex situations and interact with humans, robots that adapt to a novel situation are needed.

Classical robots, referred to as the *algorithmic approach* [1], appear to be acting intelligently whereas they are not. These robots have been programmed in such a way to enable them to have functional semantics, but this approach to solving problems is inadequate for the kinds of problems that need to be solved for general intelligence tasks, as this type of tasks involve some cognitive capabilities.

A *Generalist robot* is a type of robot that acts goal-oriented by making plans, actively explore its environment, and identify opportunities for actions. And it is able to solve general tasks problems due to its cognitive capabilities; in fact, generalist robots is a subset of cognitive robots [2] which are focused on practical problems limited to physical situations (e.g. search and pick an object).

Symbols need to be coupled to objects or events, to which the robot can attach meaning. And this means coupling those symbols to information received from its sub-symbolic subsystem. This sensor(actuator)-to-symbol problem is named as *Anchoring*

¹e-mail: descudero@iri.upc.edu

Problem [3][4] and it should be always considered when working with generalist robots. Moreover, from our point of view, a robot system that fulfill previous requirements but it is not able to learn for establishing this sensor-data and symbols connection could not be considered as a generalist robot. An example of that is when the relation between symbols and percepts is pre-programmed.

2. Problem statement

The Anchoring Problem (AP) can be defined as the interpretation of sensor/actuator data in terms of symbols, and the interpretation of high-level abstract reasoning results (such as the operator in a plan) in terms of real-time control for a robotic system. For establishing this sensor-data and symbols connection, learning techniques combined with human interaction are necessary. The process to solve AP is named *anchoring*, or perceptual anchoring, and it is the process of creating and maintaining the correspondence between symbols and percepts (sensor-data) that refer to the same physical objects. This is a complex problem and most of the solutions available thus far are ad hoc based on small-scale empirical studies, largely carried out in simulated domains.

An anchoring framework is defined as a three layer system (see Figure 2):

- Symbolic subsystem: symbols, predicates and an inference mechanism.
- Predicate grounding relation
- Sub-symbolic subsystem: percepts and attributes.

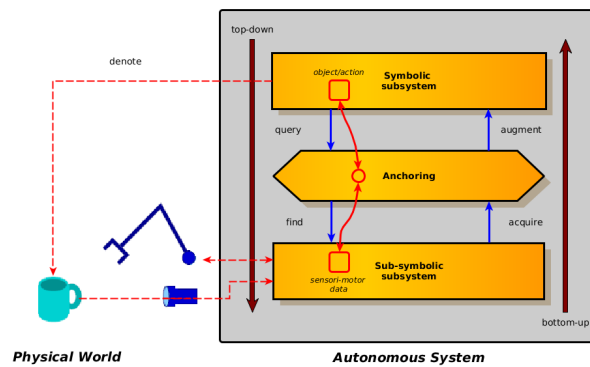


Figure 1. Anchoring framework and functionalities

The lower area is the sub-symbolic (or sensori-motor) subsystem which contains percepts and attributes. Percepts are collections of measurements originated from the same object while attributes are measurable properties from the percepts. On the top of the framework there is the symbol subsystem, it is composed by a set of symbols, predicate symbols and an inference mechanism. The symbols denote objects while the predicate symbols describe symbolic properties of the objects. Between the symbol and sub-symbolic subsystems, there is the predicate grounding relation; it embodies the correspondence between the predicates and the attributes. That is, it associates properties at sub-symbolic level to properties at symbolic level, through structures called anchors.

For manipulating anchors four abstract functionalities are defined: *acquire*, *augment*, *find* and *query*. The first two functionalities, acquire and augment, are used to create an anchor requested by the sub-symbolic subsystem. This subsystem detects a new object and request to the system to anchor this percept to the corresponding symbol in the symbolic area. And the last two, find and query, are used by the system to anchor a symbol requested by the symbolic subsystem to a percept in the sub-symbolic subsystem, in case it was not previously anchored.

These functionalities could also be classified based on how anchors are created: top-down (query and find) and bottom-up (acquire and augment). Essentially, the terms top-down and bottom-up represent the two ways of traversing within the anchoring space, semantically and perceptually, respectively. In a *top-down* approach, the AP becomes that of aligning symbols to feature sets in the robot’s perceptual stream, whereas *bottom-up* approach employs some models of sensor data interpretation in order to identify relevant properties of the world.

For us, the AP includes as well all the symbols that a robot needs to manage for executing an activity in the real world. It makes sense to anchor the symbolic concepts of robot motor actions to actual physical movements of the robot. Other authors (e.g. [2]) have extended the problem to add actions, too, as we do.

Table 1. Learning techniques depending on anchor type (what) and anchoring method (how)

| | Bottom-up | Top-down |
|----------------|------------------------------------|---------------------------------------|
| Objects | Unsupervised learning | Knowledge-base Supervised learning |
| Actions | Reinforcement learning Coaching | Imitation learning |

The learning techniques used to create anchors (Table 1) depend on what we are trying to anchor (objects or actions) and how we are doing it (top-down or bottom-up); some learning techniques will be more suitable than others.

3. Description of the system

Our system to work in the AP is based on ROSPlan [5], which is responsible of the high-level representation. It was designed (see Figure 2) after a deep analysis of the problem and considering how to apply learning and HRI techniques that are needed for the anchoring process.

3.1. Symbolic area and ROSPlan

In the AP a symbolic area must exist. This area is defined by ROSPlan in our system. ROSPlan is a framework that links two standards, PDDL2.1 and ROS, by encapsulating planning, action dispatching and knowledge gathering. Main components of ROSPlan are the Planning System and the Knowledge Base which are ROS nodes.

The sensor data is continuously gathered, parsed from sensors and passed to the Knowledge Base through specific ROS services. Information stored in the Knowledge Base is used to construct planning problem instances. The actions executed by the Planning System are dispatched to lower-level controllers as ROS actions, and they respond reactively to immediate events and provide feedback.

3.2. Sub-symbolic area and sensori-motor module

The sub-symbolic area is mapped to the sensori-motor module in our system. This module is composed by the sensors, the actuators, and the low-level controller, which is defined as a set of ROS nodes related to perceptual and manipulation algorithms that acquire percepts from the environment or apply actions to the robot, respectively.

We need to characterize objects (e.g. cup, box) and trajectories that represent an action (e.g. simple movements and pick/place). To characterize objects we extract classical features such as color and geometrical properties, while to characterize trajectories we use Dynamical Movement Primitives (DMPs) encoding positions of the joints.

3.3. Anchoring and learning

The anchoring module connects ROSPlan and the sensori-motor module. In Figure 2 the four abstract functionalities for manipulating anchors (acquire, augment, find and query) are shown in our system. In this module, the link between symbols and sensori-motor information is done based on learning techniques and interaction with humans.

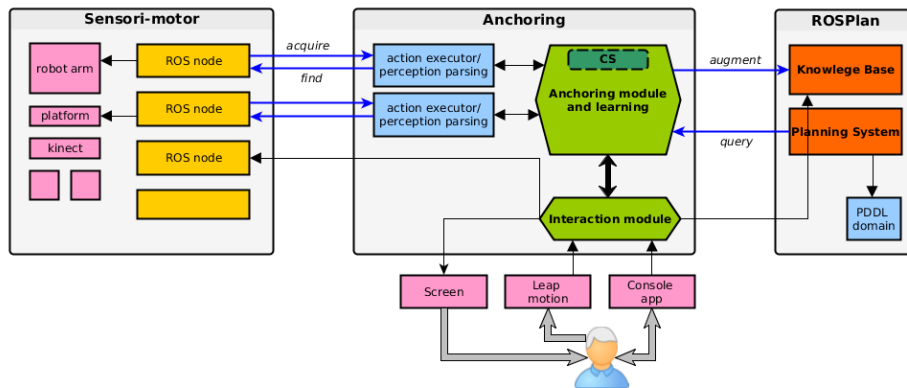


Figure 2. Our anchoring system with ROSPlan

Anchoring of an object is similar to work done in other areas as cognitive vision and semantic image interpretation. However, instead of getting percepts only for cameras, non-vision sensors (e.g. touching sensors) may be used as well. Anchoring actions is done by applying learning techniques that involve actuators in addition to perceptions; these approaches could be classified in learning low-level motion trajectories, learning high-level tasks, and refining a learned task.

In our system we use *conceptual spaces* [6] inside the anchoring module. Conceptual spaces (CS) provide a geometric treatment of knowledge which bridges the gap between the symbolic and sub-symbolic approaches.

The system is defined to be always running and the user is only able to interact with it in two ways, either through ROSPlan by setting a goal, a fact or doing a query, or through the interaction module, which is the one responsible of managing the interactions between the user and the system, specially when the user is answering questions to the robot or teaching it some actions. The way to interact with the robot depends on the

interfaces that it provides, too. In our system, we plan to use gestures to answer simple questions/requests and a console application to add knowledge (e.g. facts) or to request a task. On the other hand, the robot will show us information related to its questions through a screen mounted on it. Another way that the user will interact with the robot is by hand: he/she will teach actions using kinesthetic learning by demonstration and correct some learned ones.

Some situations in which anchoring must be done when the system is running are:

- the user requests a goal that is using one or more symbols that are new to the robot, then the robot requests help during plan execution;
- symbol encoding for high-level representations (e.g. sequences of predefined actions) using learning by demonstration [7] [8];
- assignment of new information in the sensori-motor space (e.g. joint angles and images) to a symbol when the robot requests it;
- refinement of a learned skill when the robot it is trying to solve a task and it is realized that this anchor was incorrectly done.

4. An initial experiment

The first experiment to be done is based on the ConSCIS architecture created by Chella et al. [9], where they demonstrated the ability of a robotic arm to learn and imitate a set of movement primitives acquired through a vision system. Our objective is to study how their ideas can be mapped to our system based on ROSPlan and PDDL files. Other authors have also used some of Chella's ideas to test their proposed architecture [8].

Although our experiment is based on the previous one, we would like that our robot (Figure 3-a) was able to sort some videotapes placed on a table by putting the ones of the same color in a stack (only two different colors are used), see Figure 3-b. The robot system is composed by a Kuka LWR robot with a gripper and a smart camera mounted on it. As our system is based on ROSPlan, before starting the experiment we have to: 1) define the PDDL domain file including actions (e.g. *search-object(o)* and *place-on(o, p)*); 2) create ROS nodes that map PDDL actions to robot services (e.g. smart camera and arm controller); and 3) update the Knowledge Base with needed information.

The robot system will be simulated in MORSE [10] as it is not strictly needed to use a real robot for this experiment.

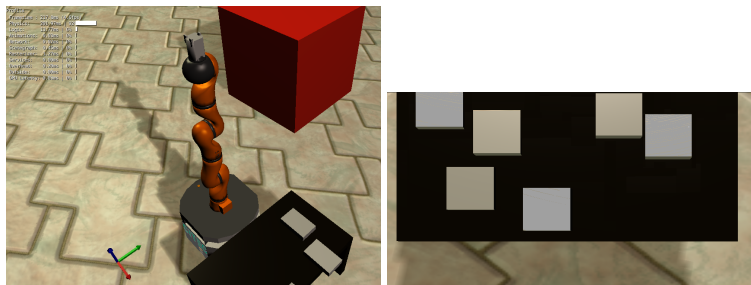


Figure 3. Setup of the initial experiment (a) the robot system, and (b) a image showing a random distribution of the videotapes on the table

In Chella's experiment, the learning of concepts is bottom-up, and it is done applying clustering techniques over three conceptual spaces: perceptual, situation and action. We plan to work on both approaches of anchoring, top-down and bottom-up. We are going to use conceptual spaces, too; and the learning/interaction processes to be applied are: 1) to anchor percepts to videotape by robot request (bottom-up); 2) to infer object type of videotapes through Knowledge Base (top-down); and 3) to learn pick-object action to be used with a videotape by robot tele-operation (top-down).

5. Conclusions and future work

In this paper we have presented the system we are creating to work on the study of the Anchoring Problem in generalist robots. This system was designed considering learning and HRI techniques as they are necessary for the anchoring process, and it is based on ROSPlan, which is responsible of the high-level symbolic representation.

ROSplan will be extended with new mechanisms that will be added as external ROS nodes and applications. These mechanisms could be grouped as: anchoring processes, learning techniques, and human interaction helpers. This high-level objective could be split in two more concrete ones: (1) to link an existing symbol with a learned action by interaction; and (2) automated code generation of ad-hoc ROS nodes that connect symbols to specific perceptions/actions.

Before beginning with our initial experiment, we have to complete the development of the system we are proposing. We have to prepare ROS nodes that represent actions defined in the PDDL domain file and add interfaces to interact with the robot. After that, we will work on the execution of our experiment, to anchor actions based on learning by demonstration and reinforcement learning.

Acknowledgements: This work was partially funded by Spanish research project DPI2013-42458-P.

References

- [1] U. Kurup and C. Lebiere *What can cognitive architectures do for robotics?*, Biologically Inspired Cognitive Architectures, vol. 2, 2012.
- [2] M. Daoutis, S. Coradeschi, and A. Loutfi, *Towards concept anchoring for cognitive robots*, Intelligent Service Robotics, vol. 5, no. 4, 2012.
- [3] S. Coradeschi and A. Saffiotti, *An Introduction to the Anchoring Problem*, Robotics and Autonomous Systems, vol. 43, no. 2-3, 2003.
- [4] S. Coradeschi, A. Loutfi, and B. Wrede, *A Short Review of Symbol Grounding in Robotic and Intelligent Systems*, KI - Künstliche Intelligenz, vol. 27-2, Springer, 2013.
- [5] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtós and M. Carreras, *ROSPlan: Planning in the robot operating system*, 25th ICAPS, 2015.
- [6] F. Zenker and P. Gärdenfors *Applications of Conceptual Spaces: The Case for Geometric Knowledge Representation*, Synthese Library, vol. 359, Springer, 2015.
- [7] A. Billard, S. Calinon, R. Dillmann, S. Schaal, *Robot programming by demonstration* Chapter 59 in Handbook of Robotics, Springer, 2008.
- [8] R. Cubek and W. Ertel, *Learning and Application of High-Level Concepts with Conceptual Spaces and PDDL*, PAL 2011, 21th International Conference on Automated Planning and Scheduling, 2011.
- [9] A. Chella, H. Dindo, and I. Infantino, *Imitation learning and anchoring through conceptual spaces* Applied Artificial Intelligence, vol. 21, no. 4-5, 2007.
- [10] S. Lemaignan, G. Echeverria, M. Karg, J. Mainprice, A. Kirsch, and R. Alami, *Human-robot interaction in the MORSE simulator*, ACM/IEEE International Conference on Human-Robot Interaction, 2012.