

Research Article

Open Access

Alexandre Angleraud*, Quentin Houbre, and Roel Pieters

Teaching semantics and skills for human-robot collaboration

<https://doi.org/10.1515/pjbr-2019-0025>

Received February 10, 2019; accepted July 23, 2019

Abstract: Recent advances in robotics allow for collaboration between humans and machines in performing tasks at home or in industrial settings without harming the life of the user. While humans can easily adapt to each other and work in team, it is not as trivial for robots. In their case, interaction skills typically come at the cost of extensive programming and teaching. Besides, understanding the semantics of a task is necessary to work efficiently and react to changes in the task execution process. As a result, in order to achieve seamless collaboration, appropriate reasoning, learning skills and interaction capabilities are needed. For us humans, a cornerstone of our communication is language that we use to teach, coordinate and communicate. In this paper we thus propose a system allowing (i) to teach new action semantics based on the already available knowledge and (ii) to use natural language communication to resolve ambiguities that could arise while giving commands to the robot. Reasoning then allows new skills to be performed either autonomously or in collaboration with a human. Teaching occurs through a web application and motions are learned with physical demonstration of the robotic arm. We demonstrate the utility of our system in two scenarios and reflect upon the challenges that it introduces.

Keywords: human-robot interaction, cognitive architecture, knowledge representation and reasoning, symbol grounding, semiotics

1 Introduction

Industry and academia are advancing in the development of applications with collaborative robots (cobots) in industrial settings or social contexts [1, 2]. In a scenario where robot and human share the same workspace, the ability from both parties to adapt their behavior to each other is required. Furthermore, it is expected from humans workers that they would be able to adapt to new environments they could be confronted to. In a similar way, a robot can be programmed to have general and domain-specific knowledge such as names of the main techniques, mechanisms implied in the task or the tools that are used. However, after starting its operations, it should also be capable of acquiring what is referred to as procedural knowledge. This category includes information related to the specific execution of a given skill or the division of workload. That said, actuating a robot is not a trivial task and programming such platforms would require a high-level computer science expertise. As a result, to successfully deploy a robot, it needs to be delivered with an intuitive interface for interaction and have access to learning capabilities to enable its coworkers to teach it the necessary operational skills.

Current state of the art collaborative robots come with such systems (e.g., Franka Emika's Panda [3], Rethink Robotics' Sawyer [4]). They are delivered with a library of built-in apps that allow for the end-user to program various sequences of motions and grasping tasks, without having to go through a development phase, which considerably speeds up getting started with the robot. However, by doing so, the semantic information about the robot actions is lost. This means that any kind of collaboration with the robot in the sense of performing a task together is hardly thinkable.

In addition, a key element in human teams is natural language, which eventually consists in a symbol system. Hence it carries out properties resulting from years of studies in other scientific fields such as philosophy and cognitive science. Despite being less popular in the robotics field, we see it as crucial to achieve seamless human-robot collaboration, as it brings a way to build a mutual under-

*Corresponding Author: Alexandre Angleraud: Automation Technology and Mechanical Engineering, Cognitive Robotics, Tampere University, 33720 Tampere, Finland; E-mail: alexandre.angleraud@tuni.fi

Quentin Houbre, Roel Pieters: Automation Technology and Mechanical Engineering, Cognitive Robotics, Tampere University, 33720 Tampere, Finland; E-mail: quentin.houbre@tuni.fi, roel.pieters@tuni.fi

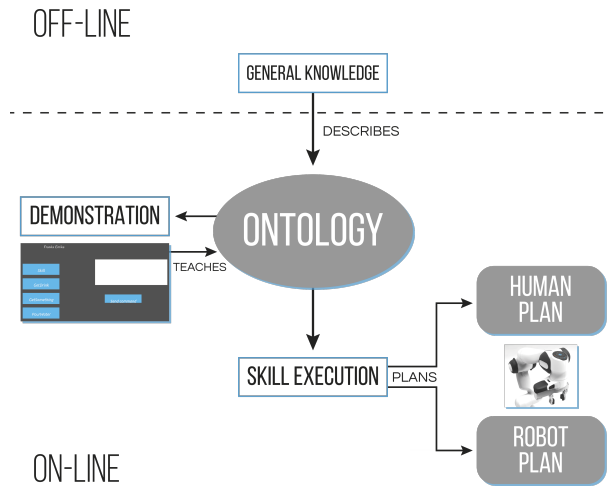


Figure 1: General organization of the collaborative platform. Skill description occurs off-line and independently of the robot used. Operative knowledge is then demonstrated on-line after which the skills can be activated by natural language commands or through the GUI.

standing of the world and the tasks to be performed. In this work, we thus propose a system that integrates a symbol system to develop learning and reasoning features. It is used (i) to provide a way to teach new skills to a robot and (ii) as a base for a grounding mechanism to solve communication issues during the interaction. These developments are build upon previous work presented in [5] that demonstrated a proof of concept of our symbol manipulation approach in human-robot communication. This work extends this approach and our contributions are:

1. Extension of the knowledge base for robot and task capabilities
2. Integration of semantics and skills in one framework for shared task planning and execution
3. Two example tasks to demonstrate the capabilities of the system: service and industrial

To teach new robot skills two interfaces are available. On one hand a web based platform to teach in advance the general knowledge required for performing a task, and on the other hand a GUI to teach the operative knowledge. The reasoning deployed on the robot consists of three layers of processing in which the robot can sense, think and act. Finally, a set of ontologies comes as the central component. These ontologies are nothing less than a conceptual representation of the robot's knowledge. This representation then allows for automated planning strategies to be adopted. We demonstrate the system with two simple but realistic proof of concept scenarios in which an operator

has defined the skills required and human and robot can agree together on how to collaborate before performing the shared task (see Figure 1).

This paper is organized as follows. We first introduce its core ideas through some related work in Section 2, after which our model for communication and reasoning is defined conceptually in Section 3. Section 4 explains in detail the choices for the presented system and Section 5 presents demonstrations of the system together with a discussion on its current limits and next challenges.

2 Related work

We review here some of the (many) studies around knowledge representation and organization that can be found in the literature. However, before doing so, it is important to consider a few definitions. Broadly speaking knowledge will here refer to everything that the user or the robot believe as true [6] (as opposed to other fields in which it could refer to everything that has been verified as true). A distinction is also made between beliefs and knowledge as the former is more personal and might influence one's perspective on the latter [7]. This information can be further subdivided into declarative, procedural and conditional [8]. In Section 2.1 we focus on the transfer process from human to robot of the declarative knowledge. Furthermore, it is often expected of a robotic system to be able to verbalize this knowledge and hence we then review in Section 2.2 the strategies around the symbol grounding problem or how to connect the information to the syntax of human language. Finally, Section 2.3 introduces methods for automated planning using the semantic information available about a skill.

2.1 Semantics of a skill

Robots for social interaction or industry require knowledge of the world in order to act in it. Knowledge that can be taught to a robot, for example, how to perceive the environment or how to perform a task. This applies for common sense knowledge [9, 10] or industrial skills [11, 12]. Transferring our knowledge to robots has been an open problem for many decades as what we consider as common sense becomes quickly very complex to represent. Every action taken has a chain of consequences that need to be considered, even when dealing with a single robot acting alone in the world (see for example the "midnight snack problem" [13]).

In Human-Robot Interaction (HRI), cognitive mechanisms responsible for the emergence of meaning in language are explored [14] with questions such as, how to represent knowledge in such a way that it explains the capacity to discuss about perceived objects for example. Recent projects such as [15] address this problem under the term of data fusion and [16] underlies the cognitive functionalities that could allow the meaning of words to directly emerge from the sensory feedback. Doing so it introduces the field of Symbol Emergence in Robotics (SER).

Other approaches than this conceptual modelling include neural networks as in [17] where Long Short Term Memory networks are used to model language. Alternatively, an opposite approach is suggested in [18], which claims that no representation whatsoever is needed and that the intelligent behavior should emerge from the lower levels of the reasoning chain, staying then closer to the raw sensory information. Following this bottom-up philosophy, [19] shows how self exploration and social interactions can contribute to language acquisition while [20, 21] are devoted to understanding and modelling biological mechanisms.

Finally, a great deal of research has focused on building a bridge between what the robot can perceive and its representation of the world. For example, perception of geometrical properties in a scene [22] or perception of inferred human activities [23]. Additionally, [24] explored how a robot can model by itself the planning problem (in a symbolic fashion) from the sensory information.

2.2 The symbol grounding problem

When representing the world using semiotics [25], the main task is the connection between a symbol and how it should be interpreted. Harnad defines in [26] what he calls the symbol grounding problem, opening the door for numerous projects investigating how to teach knowledge to a robot. For example, how to pick up an object on the fly while interacting directly with the robot [27] or how to teach action composition through dialogue [28]. First steps have even been made to acquire the notion of context while interacting with the robot [29]. Working the other way around, [30] teaches a robot to represent image concepts in a way that allows it to describe them.

As it is impossible to program in advance everything the robot needs to know, the symbol grounding process is a convenient way to increase the autonomy of the robot by making it able to solve ambiguities and misunderstandings. In previous work we have developed such system [5] which allows to handle unknown concepts through sym-

bol manipulation based on natural language interactions. Given that the user would respect a certain pattern for communication the robot could make new connections using its current knowledge and learn new concepts. High-level concepts such as 'cook pasta' where taken to demonstrate knowledge transfer from human to robot and independent knowledge acquisition through reasoning.

2.3 From interaction to collaboration

Complex systems aiming at integrating knowledge representation to task planning have been previously developed so that the robot is aware of its environment and reacts to events [11, 31]. However, to make robots capable to enter our every day life we need to make them as simple as possible [32]. In industry where task programming or demonstration typically requires a lot of expertise and time, methods are needed to teach new skills in an intuitive way [33].

In [34] for example, simple programming is allowed in the form of setting program parameters using spatial augmented reality for visualization and a touch-enabled table and robotic arms as input devices. [1] strives to develop a collaborative robot from the starting point of a mobile manipulator and focuses on creating a human-robot interface with intuitive interaction not only in the programming phase but also during the general operation of the robot.

In any case, robots should be able to modify their behavior according to the situation. Adaptation can occur at the communication level [35], or at the action level [36, 37] with the robot taking initiative to best assist the user.

In [38] a system referred to as FlexHRC, for flexible human-robot cooperation, is proposed. It relies on wearable sensors for human action recognition, graphs for the representation and reasoning upon human-robot cooperation on-line, and a task priority framework to decouple action planning from robot motion planning and control. This system thus focuses on a real environment. Another approach uses technologies such as virtual reality glasses to provide a virtual environment for interaction with the robot. This is shown in [39] where a task modeling approach is proposed to build a conceptual model of the entire work sequence in a hierarchical "state and transition" definition sequence.

What we propose in this paper is a hybrid, skill-centered way of teaching, combining a web based user interface to describe the general knowledge required to perform a certain task with a graphical user interface that allows the user to customize the execution through demonstrations. We evaluate and demonstrate the approach with

two experiments and thus two different ways of performing an action. The first one assigns the robot to work in autonomy and the second one assigns the robot to work in cooperation with a human, according to a balanced workload.

3 Language model for collaboration

For successful human-robot collaboration the development of the robot should be open-ended, i.e., even after being deployed it should still be able to modify its behavior and learn new tasks. In this paper, the robot is considered a passive learner and as such, can only receive new information from the user. The most natural way for humans to communicate is through natural language, or in other words exchanging meaningful symbols representing the environment. In [14] Jackendoff identifies the main categories these symbols can belong to. The following section aims to explain how they have influenced our language model. 3.1.1 discusses the [EVENT] category, 3.1.2 describes how we adapted [ACTION], and 3.1.3 presents the interpretation of [THING] and [PLACE].

3.1 Semantic memory

3.1.1 Requests uttered by the user

The considered [EVENT] are the requests delivered to the robot, i.e., the set of commands that a user is able to send. A **Request** R is defined as :

$$R = (A, T), \quad (1)$$

where A refers to the symbol of an action and T refers to the symbol of a target.

3.1.2 Symbols related to the robot skills

The [ACTION] component of a request corresponds to the potential capabilities of the robot such as performing a motion, or speak out a question. At the lowest level, an action is composed of an action primitive and an optional target to act on. To create more useful bricks they are nested inside each other to form an **Action** A with a number N of sub-actions such that:

$$A = \{a_i(t_i)\}_{i=1}^N \quad (2)$$

where a_i and t_i are the sub-action and sub-target of the step i for the action A .

To organize the robot knowledge we introduce the notions of **Task** and **Step**. A **Task** is a logical set of actions directly related to the physical capabilities of the robot:

$$K = \{a_i\}_{i=1}^N \quad (3)$$

where a_i is the action i for the task K .

While a **Step** is an arbitrary group of N tasks defined by the user for the purpose of dividing the work during the execution of joint tasks:

$$S = \{k_i\}_{i=1}^N \quad (4)$$

where k_i is the task i for the step S .

3.1.3 Symbols related to physical entities

[THING] and [PLACE] are notions of concrete elements from the environment. They are represented by the **Target** T component of a request and are associated to a number N of properties, thus:

$$T = \{p_i\}_{i=1}^N \quad (5)$$

where p is a property of the target attached to a primitive data type value (a string or an integer for instance).

3.2 Abstraction and generalization

We, humans, talk about our actions with different granularities in our every day speech. Instructions such as "Serve water" and "Get a glass; Pour water in it" eventually describe the same situation but at a different level of *abstraction*.

Furthermore, to be usable the grounding system also needs *generalization*. This will allow it to use previous knowledge to attribute semantics to new symbols used in a natural language command without having to manually define their meaning.

Based on previous works [5] we use interactive communication to endow the system with those two abilities. Hence, in our framework, after being confronted to new symbols the robot will engage a dialog with the human. If trying to use higher level instructions, its meaning can be attached to the lower level components it is composed of. Whereas, if trying to use different parameters for an instruction already known, the semantics of the different parameters can be linked together.

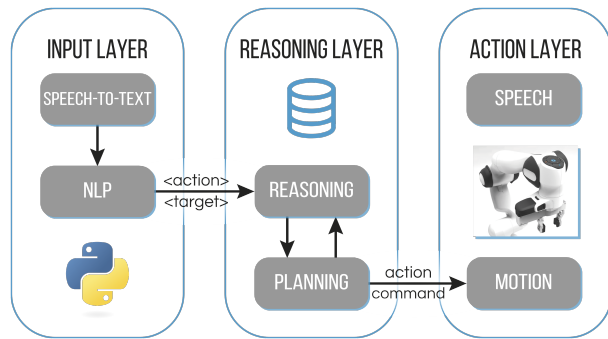


Figure 2: The 3 layers approach allowing the robot to sense, think and act. The first layer listens for audio input from the environment. The second layer reasons about available skills and establishes action plans. Finally, the action layer handles the physical capabilities implementations [5].

4 Awareness design for a collaborative robot

The software architecture is twofold. On one hand a web based application allows an operator to define skills for the robot. And on the other hand a ROS package that actuates the robot featuring the following: a rqt plugin acting as the robot graphical user interface (GUI) connected to it through ROS, a module that handles the communication between the human and the robot and a reasoning module responsible for task planning and decision making. 4.1 will present how to use knowledge in our paradigm while 4.2 will discuss how a user can add knowledge into the system.

4.1 High level cognitive capabilities

Aside from the teaching processes, the reasoning system is composed of 3 layers (see Figure 2). The interaction module, the reasoning module and the action module. These are based on a previous iteration of our work and explained in detail in [5]. The interaction module has two functionalities available. First, it enables sending requests through natural language providing an intuitive alternative to a more traditional GUI. It was decided to allow both speech and written commands as the work environment might be different from one industry floor to another which might impose different constraints to the workers. Second, it triggers the grounding process when an ambiguity is detected.

4.1.1 Conceptual reasoning

This module converts the audio input from the user into a string suitable for processing. A set of rules is then applied to the string and outputs an action and a target [5]. The set of rules will differ according to which state the robot is in (see Figure 3). After the analysis of the string, an action and a target are chosen and the two symbols are transmitted to the reasoning module.

The reasoning module contains a set of SPARQL queries which allow for reasoning by interacting with the knowledge base and retrieve various information about the robot skills. The planning can be done with as a goal, performing the task in autonomy, letting the human perform everything or adopting a certain collaboration. In the last case, the division can only be done given one criteria, trying to have a good balance between the workload of the human and the robot. In this work the planning process is done automatically by the system. Note that before being able to reason about the skills of the robot, it is necessary to process the information present in the knowledge base. To this end, we developed a parser which takes the skill definition previously made and translates it into python objects usable by the reasoner.

The last module, the action module, takes into account the physical abilities of the robot. Therefore, this part depends entirely on the platform.

Algorithm 1 describes how a user request is processed. First, Line 2 extracts two symbols from the request sent by the user. An action and a target that are then assessed in Lines 3 and 4 to identify the corresponding skill in the knowledge base. If a skill is found, the details required to plan for the execution are retrieved Line 5. However, if no skill is found depending on which symbol caused the search to fail the robot can require more information about it in Line 9 for an action or Line 13 for a target.

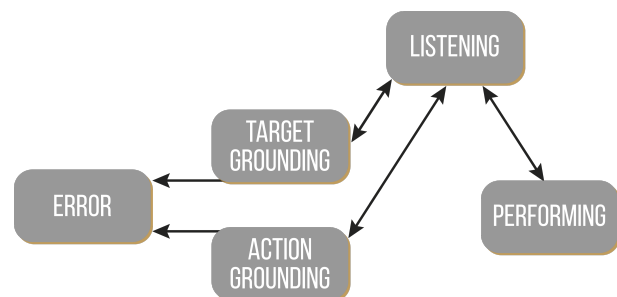


Figure 3: State machine of the Speech Recognition module. Each state has its own rules that guide the type of sentences that the robot expects to hear.

Algorithm 1: Match user request with available skills

```

(1) func process(request) ≡
(2)   action, target := pattern_matching(request)
(3)   set_skills := check_syntax(action, target)
(4)   skill := check_semantic(action, target, set_skills)
(5)   steps, constraints := retrieve_actions(skill)
(6)   call build_graph(steps, constraints)
(7)   if RequestSyntaxError
(8)   then
(9)     request_skill_explanation()
(11)  if TargetUnknownError
(12)  then
(13)    request_target_explanation()

```

4.1.2 From skill definition to action plan

Let us take a closer look at two functions implemented in the reasoning module to translate the conceptual knowledge into an executable plan for the robot. First, the symbolic knowledge is used to build the action graph of a skill, as described in Algorithm 2.

Algorithm 2: Create action plan graph

```

(1) func build_graph(steps, constraints) ≡
(2)   G := Graph()
(3)   working_agent := 'Human'
(4)   for i := first_step to last_step do
(5)     t_i := retrieve_tasks(i)
(6)     g_i := init_generator(t_i)
(7)     add_node(i, g_i, G)
(8)   for j := first_constraint to last_constraint do
(9)     add_edge(j, G)
(10)  attribute_node(j[1], working_agent, G)
(11)  switch_working_agent()

```

After initializing the structure to hold the action graph, the algorithm loops first through all the step and then through all the constraints. For each step Line 5 retrieves the information about what is supposed to be done. Line 6 then initializes an iterator that will be used to send the action commands during the execution. Finally, this information is embedded in a node that is added in the action graph (Line 7). For each constraint Line 9 adds the corresponding edge in the graph. For now no specific rea-

soning is done regarding the task attribution and Lines 10, 11 make sure that an equivalent number of tasks are attributed to the user and the robot.

Second, Algorithm 3 shows how the robot decides what should be done next.

Algorithm 3: Navigate graph to decide the next action

```

(1) func decide_next_move(current_step) ≡
(2)   if agent_of(current_step) == 'Human'
(3)   then
(4)     exit
(5)   else
(6)     current_step.generator.next()
(7)     if StopIteration
(8)     then
(9)       call find_next_step(current_step)
(10)  where
(11) func find_next_step(current_step) ≡
(12)   mark_as_complete(current_step)
(13)   for i := first_edge to last_edge do
(14)     if i[0] == current_step
(15)     then
(16)       current_step := i[1]

```

If Line 2 detects the current action to be done by the human, the algorithm exits. However, if the robot is executing a step, the corresponding generator will return the next move. Line 6 catches the case when the generator has reached the last move corresponding to the current step. Line 12 updates the graph to indicate that this step has been completed and then iterates through the edges to find the successor which becomes the new current step (Line 16).

4.2 Interactive learning

Teaching declarative knowledge to the robot can be done via a web based app (see Figure 4). This allows for some flexibility as it is not linked to a specific platform. The goal is to provide the robot with sufficient knowledge to be included in a team on the field but leave space for workers to tailor it to their working habits. Through the interface, one can also examine the knowledge base and query it if necessary.

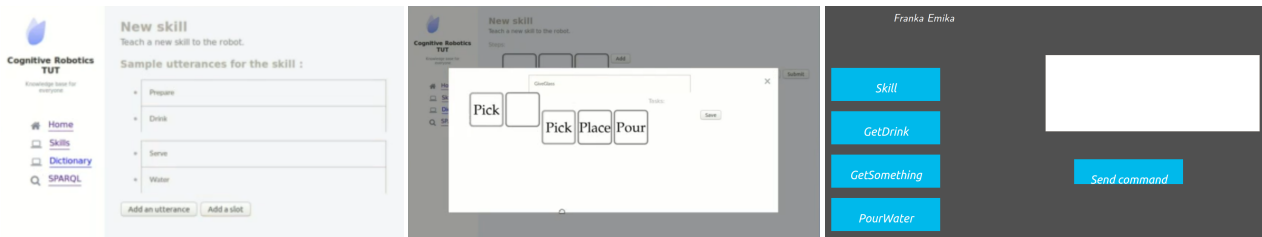


Figure 4: Graphical user interface. Left: Off-line web app allowing to describe new skills to the robot. Here the language syntax is defined. Center: While here are specified the sub-actions. Right: Natural language requests can be sent via the GUI.

4.2.1 Teaching skills

The main feature of the app is the definition of a new skill. In practice this means adding the semantic information about a skill in the knowledge base using ontologies for representation. This includes a name for the skill, how to activate it and the different components that constitute it. For example, the actions required for pouring water in a glass.

The teaching process is the following. First, the name of the skill has to be defined. A common practice for naming is to use verbs that explicitly state the occurring actions. The second step consists in choosing the set of occurrences that trigger the skill being defined. In similar ways as with devices such as Alexa (the virtual assistant developed by Amazon), one can indicate a set of <action>, <target> pairs that, upon reasoning would activate the planning of the skill. Finally, the user needs to define the skill itself and its properties.

It was decided to define in advance a set of tasks that constitutes the poll to choose from, these are: picking, grasping, pouring. The user has to define steps and for each of them associate at least one action. The subdivision of the steps is important as they will be considered as atomic pieces of the skill from the robot's perspective. This means that a step will not be interrupted and the collaboration occurs one level higher. Every task included in one step will therefore have to be performed by the same agent. Once these steps are completed, the ontology corresponding to the new skill will be generated (see Figure 4) and is immediately available in the knowledge base. This implies that the skill is directly available for demonstration to a robot.

4.2.2 Solving ambiguities

When the user sends commands to the robot using natural language, the terms used can sometimes be ambiguous for the robot. Algorithm 4 describes the strategy to resolve the

conflict when this occurs for an action symbol and Algorithm 5 when this occurs for a target symbol.

Algorithm 4: Register new link between uttered skill and available knowledge

```

(1) func grounding_action(tokens)  $\equiv$ 
(2)   valid := matching_attribute(tokens)
(3)   if valid == True
(4)   then
(5)     pattern_matching(tokens)
(6)     if mode := 'building'
(7)     then
(8)       request_definition(new_skill)
(9)       if mode := 'relating'
(10)      then
(11)        relating_skills(new_skill, related_skill)
(12)   new_state := "listening"

```

In Algorithm 4, Line 2 makes sure that the user is using a skill already known to explain the new symbols. If yes then Line 7 separates the symbols and identifies the kind of grounding being used. Two skill grounding cases are considered. It can be a definition from scratch, in this case the user defines a high level concept with a set of sub tasks and Line 8 will register the sub task mentioned and confirm to the user that he or she can continue explaining. The second case is the user relating the new symbols to a skill already known (Line 11). Finally Line 12 brings the internal state machine of the robot back to the Listening mode.

Similarly as in Algorithm 4, Line 2 of Algorithm 5 compares the terms used to explain the entity with the content of the knowledge base. The pattern matching rules corresponding to physical entities grounding are applied (Line 7) to extract the symbols and the grounding case. Line 10 considers the definition of the symbol using a higher level concept while Line 14 considers relating the new symbol

Algorithm 5: Register new link between uttered entity and available knowledge

```

(1) funct grounding_target(tokens) ≡
(2)   valid := matching_attribute(tokens)
(3)   if valid == True
(4)   then
(5)     mode, target, known_entity :=
(7)     pattern_matching(tokens)
(8)     if mode := 'sub_class'
(9)     then
(10)      define_sub_class(target, known_entity)
(12)     if mode := 'equivalent'
(13)     then
(14)      define_equivalent(target, known_entity)
(15)   new_state := "listening"

```

to an equivalent concept. Line 15 again brings the internal state machine of the robot back to the listening state.

4.3 Teaching operative knowledge

Teaching the operative knowledge to the robot can be done through a GUI developed as a rqt module (see Figure 4) and thus connected to the robot through ROS. As a reminder, the goal of this interface is for the end user of the robot to be able to receive the general knowledge about the available skills and add specific information as to how it should be performed. In other words, the user can demonstrate to the robot the different steps of a task.

The interface offers two choices. First, a list of skills will be displayed as buttons. Clicking on one of them will lead to the respective interface of the skill in which each step is retrieved, as well as the tasks that are associated with it. The operator can then demonstrate to the robot how to perform each task and execute it. This information is not stored in the knowledge base but rather on the robot itself. Second, if the skill has previously been demonstrated already, then the user also has the choice to simply input the natural language commands in the text editor field.

5 Demonstrations and discussion

To demonstrate our approach two cases are considered: a service robotics task and an industrial robotics task. The experiments are conducted in a university laboratory with people familiar to robots. We use the Panda arm from

Franka Emika (see Figure 7), a robot specially designed for collaborative tasks. Our communication uses the Google text-to-speech engine ¹ and the Festival speech synthesis system ². As for the action graphs in the reasoning module they are developed with the NetworkX package ³. In general, the software is developed for integration into ROS and released under open source license ⁴.

5.1 Scenarios

5.1.1 Pour water in a glass

As a first example, we considered handling simple social robotics skills where the interactive part might take more importance than the definition of the skill itself. We took here the example of pouring water in a glass (see Figure 5). A video presenting this example is also available online ⁵.

To start with, we defined the general knowledge about this skill via the web app and thus this could be done remotely from the robot itself. For the name of the skill we chose **PouringDrink**. The two following commands were given as a trigger (<Pour>, <Drink>) and (<Serve>, <Water>). Finally we chose to define this skill as a combination of three steps :

- **GetGlass** : Position the glass at a specific location (human)
- **Serve** : Pour the water in the glass (robot)
- **HandGlass** : Give the glass to the user (human)

During the procedural knowledge teaching phase, each step is demonstrated to the robot. In the current implementation it is assumed that during the next execution, the size of the glass or its position would be the same. The container for the water should also remain similar.

Typically, this kind of request could be given using many different terms. Therefore, several different triggers were tried to activate the skill. However, each user could have its own way to ask the robot. In this case the ambiguity can be solved using natural language conversation as described in Section 4.2.2 and as demonstrated below.

HUMAN: Prepare a glass.

ROBOT: I am sorry, this does not match any request.

HUMAN: I will teach you the action to prepare a glass.

¹<https://pypi.org/project/SpeechRecognition/>

²<https://cstr.ed.ac.uk/projects/festival/>

³<https://networkx.github.io/>

⁴<https://github.com/Zorrander/cogrob-tut-hri>

⁵<https://youtu.be/DchkPONQ5iE>

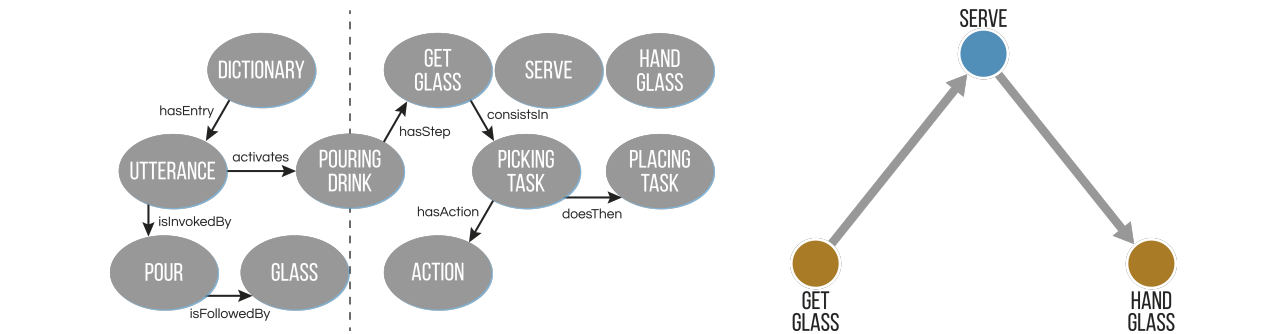


Figure 5: On the left hand is an extract of the semantic description of the skill pouring a drink (scenario 1). On the right hand is shown the plan generated by the system. In orange is what will be performed by the human and in blue what the robot is expected to do. A signal has to be sent to the robot to specify the end of the step. This can be done either by pressing a button or through natural language.

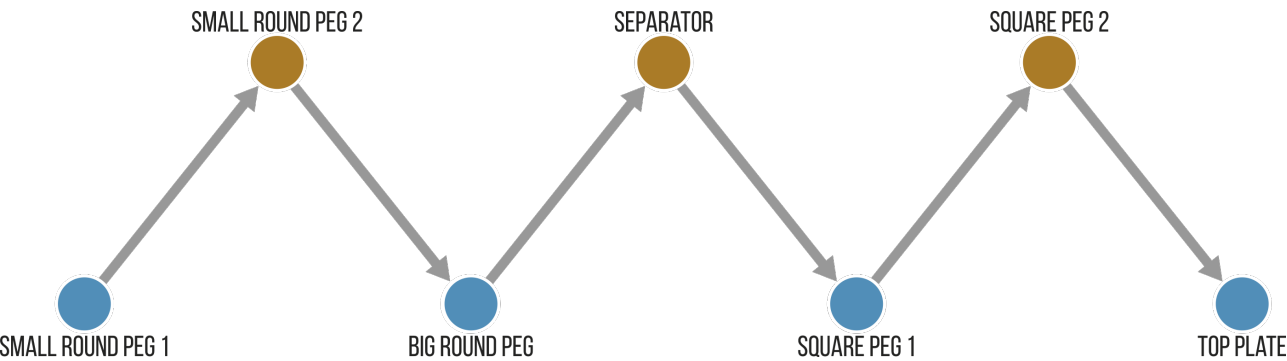


Figure 6: Plan generated by the system for scenario 2 (Cranfield benchmark assembly). In orange is what will be performed by the human and in blue what the robot is expected to do. A signal has to be sent to the robot to specify the end of the step. This can be done either by pressing a button or through natural language.



Figure 7: Top Left: Cranfield benchmark parts. Bottom left: Generated action plan (graph) and graphical user interface. Top right: demonstration of skills for human-robot collaboration. Bottom right: Robot independent execution of a task.

ROBOT: I am listening.
 HUMAN: Prepare a glass is like serve water.
 ROBOT: Understood.
 HUMAN: Prepare a glass.
 ROBOT: I will.

5.1.2 Cranfield benchmark

The second experiment (see Figure 6) simulates an industrial robotics setting and in this case assembles what is referred to as the Cranfield benchmark [40].

Before physically demonstrating the assembly task the general knowledge regarding the skill is defined as follows through the web app. The name of the skill was chosen as **AssembleBenchmark**. The following command was given as a trigger (<Assemble>,<Benchmark>). Finally, it was chosen to define this skill as an arbitrary sequence of steps, each of them composed of a pick and place task.

- **GetBackplate** : Position the backplate at a specific location (human)
- **PlaceSmallRoundPeg** : Place small round peg in the slot of the backplate (robot)
- **PlaceSmallRoundPeg** : Place small round peg in the slot of the backplate (human)
- **PlaceBigRoundPeg** : Place big round peg in the slot of the backplate (robot)
- **PlaceSeparator** : Place separator in the slot of the backplate (human)
- **PlaceSquarePeg** : Place square peg in the slot of the backplate (robot)
- **PlaceSquarePeg** : Place square peg in the slot of the backplate (human)
- **PlaceTopPlate** : Place top plate on the assembly (robot)

It is still relatively easy to define the general knowledge of this task but some limits arise when it comes to the learning by demonstration part. For large tasks this becomes quite repetitive and solutions will have to be sought in future work.

First the skill referred to as <Assemble>,<Benchmark> is shown to the robot. Once the demonstration mode activated, the human can guide the robot manually to show each step individually using the GUI to map each step to its corresponding motion. After completing the definition of the skill the human can decide to let the robot act alone or the planner can attribute steps in a balanced way between the user and the robot. In the collaborative case a signal has to be sent from the human

to the robot to specify the end of the step. This can be done either by pressing a button or through natural language.

In case the user refers to the object of the assembly with other terms, it can be solved by linking the two terms through interaction. Below is demonstrated an example of ambiguity solved by the conversational skills of the robot.

HUMAN: Assemble benchmark.
 ROBOT: I am sorry, I don't know what benchmark is.
 HUMAN: I will teach you what benchmark is.
 ROBOT: I am listening.
 HUMAN: Benchmark is like Cranfield assembly.
 ROBOT: Understood.
 HUMAN: Assemble benchmark.
 ROBOT: I will.

5.2 Limits and future work

Extending our previous developments in [5] the robot can be taught new abilities by combining a predefined set of tasks in different ways. The action space thus depends on this original knowledge. As mentioned in [13] learning mechanisms are crucial, but at least some information has to be programmed in advance for the robot to be able to learn more in autonomy. This is why it was chosen to have an initial set of basic skills that constitutes the knowledge base. Naturally, this knowledge base will extend in the future to enlarge the fields of application of our system but also to investigate its behavior when the number of skills becomes very large.

Further, while resolving ambiguities, the robot can assimilate new knowledge by creating new associations based on its prior knowledge. Interaction is the way explored in this paper to solve those situations. As such, the user is responsible for selecting which pieces of knowledge are connected. In practice, it forces the user to have at least a partial knowledge of the system but for now we limited the impact of this factor by allowing redundancy. That way, the user can create non optimal solutions but at a lower cost.

The communication relies on a rule based method using an action target complex. As of now it constitutes a constraint to authentic natural interaction and does not take into account all the particularities that can be found in the English language. As a result, for the interaction to take place, the user needs to be introduced to the communication expectations of the robot first. Nevertheless, these have the benefit of bringing predictability to the system which in an industrial setting for instance is crucial.

Since the expectations can vary in different contexts, future work will address different interfaces to our system to allow such flexibility. Moreover, it is here assumed that interactions always involve one human with one robot but user studies should then be conducted to ensure that the platform can be extended to more complex scenarios.

To achieve successful collaboration, the reasoning and planning of a skill occurs off-line and without any extensive sensory feedback. This, therefore, does not allow to modify a plan on the fly or anchor the knowledge to physical entities. In other words, it is necessary that the system would attribute steps and for the agent to behave exactly like the robot is expecting. Moreover for the execution to be successful objects of the same class should be placed in the same position have similar properties.

In this regard, further developments should include a bottom up component building on top of the current top down approach focusing on learning from demonstration by adding a suitable visual system. Being able to track separately the agent and the objects in the scene will allow the robot to detect mismatches between the original plan and the actual human behavior as well as use generic grasping policies for the different object classes, increasing the adaptability of the system. It will also allow action segmentation being the first step to a more general action grammar framework bridging the gap between the interaction and action modules.

6 Conclusion

Robots are becoming ubiquitous in our lives and they are more and more foreseen as future help to people. However, looking at human-human collaboration, for this transition to happen developing communication abilities is crucial. Not only can it be used to learn, synchronize actions and teach, but it also allows a mutual understanding of the environment between agents. This work takes our primary tool for communication, natural language, as a base to provide a system based on semiotics with built-in interaction and reasoning capabilities for teaching action semantics to robots. The proposed system gives access to a twofold teaching platform to define general and task-specific knowledge that can be shared across different platforms. In addition, a dialogue mechanism based on the conceptual reasoning allows to resolve conflicts when a user utters an ambiguous request. Two conceptual scenarios are developed and demonstrated in detail that show how to proceed with the developed platforms and the Panda arm from Franka Emika.

References

- [1] P. J. Koch, M. K. van Amstel, P. Debska, M. A. Thormann, A. J. Tetzlaff, S. Bøgh, D. Chrysostomou, A skill-based robot co-worker for industrial maintenance tasks, *Procedia Manufacturing*, 2017, 11, 83–90
- [2] M. Staffa, S. Rossi, Recommender interfaces: the more human-like, the more humans like, In: *Proceedings of the International Conference on Social Robotics*, 2016, 200–210
- [3] Panda, Franka Emika, <https://franka.de>, 2019 [Accessed: 2019-02-17]
- [4] Sawyer, Rethink Robotics, <https://www.rethinkrobotics.com/sawyer/>, 2019 [Accessed: 2019-02-17]
- [5] A. Angleraud, Q. Houbre, V. Kyrki, R. Pieters, Human-robot interactive learning architecture using ontologies and symbol manipulation, In: *Proceedings of the 27th IEEE International Symposium on Robot and Human Interactive Communication (Nanjing, China, 2018)*, 2018, 384–389
- [6] P. Ein-Dor, Taxonomies of Knowledge, In: D. G. Schwartz (Ed.), *Encyclopedia of knowledge management*, 848–855 (Idea Group Reference, Hershey, PA, 2006)
- [7] C. D. Ennis, Knowledge and beliefs underlying curricular expertise, *Quest*, 1994, 46(2), 164–175
- [8] P. A. Alexander, D. L. Schallert, V. C. Hare, Coming to terms: how researchers in learning and literacy talk about knowledge, *Review of Educational Research*, 1991, 61(3), 315
- [9] M. Tenorth, M. Beetz, KnowRob: A knowledge processing infrastructure for cognition-enabled robots, *The International Journal of Robotics Research*, 2013, 32(5), 566–590
- [10] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, M. Beetz, ORO, a knowledge management platform for cognitive architectures in robotics, In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (Taipei, Taiwan, 2010)*, 2010, 3548–3553
- [11] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, R. Alami, Artificial cognition for social human–robot interaction: An implementation, *Artificial Intelligence*, 2017, 247, 45–69
- [12] T. R. Savarimuthu, A. G. Buch, C. Schlette, N. Wantia, J. Roßmann, D. Martínez, et al., Teaching a robot the semantics of assembly tasks, *Transactions on Systems, Man, and Cybernetics: Systems*, 2018, 48(5), 670–692
- [13] D. Dennet, Cognitive wheels: the frame problem of AI, In: C. Hookway (Ed.), *Minds, Machines and Evolution*, Cambridge University Press, 1984, 129–150
- [14] R. Jackendoff, *Semantics and Cognition*, 1 edition, The MIT Press, Cambridge, United Kingdom, 1985
- [15] F. Mastrogiovanni, A. Sgorbissa, R. Zaccaria, A Distributed Architecture for Symbolic Data Fusion, In: *IJCAI*, 2007
- [16] T. Taniguchi, T. Nagai, T. Nakamura, N. Iwahashi, T. Ogata, H. Asoh, Symbol emergence in robotics: a survey, *Advanced Robotics*, 2016, 30(11), 706–728
- [17] A. Arora, H. Fiorino, D. Pellier, S. Pesty, Learning robot speech models to predict speech acts in HRI, *Paladyn, Journal of Behavioral Robotics*, 2016, 9(1), 295–306
- [18] R. A. Brooks, Intelligence without representation, *Artificial Intelligence*, 1991, 47, 139–159, DOI:10.1016/0004-3702(91)90053-M
- [19] A. Cangelosi, et al., Integration of action and language knowledge: a roadmap for developmental robotics, *IEEE Transactions*

- on Autonomous Mental Development, 2010, 2(3), 167–195
- [20] T. Ziemke, The body of knowledge: On the role of the living body in grounding embodied cognition, *Biosystems*, 2016, 148, 4–11
 - [21] C. Lyon, et al., Embodied language learning and cognitive bootstrapping: methods and design principles, *International Journal of Advanced Robotic Systems*, 2016, 13(3), 105
 - [22] L. Buoncompagni, F. Mastrogiovanni, Dialogue-based supervision and explanation of robot spatial beliefs: a software architecture perspective, In: *Proceedings of the 27th IEEE International Symposium on Robot and Human Interactive Communication* (Nanjing, China, 2018), 977–984
 - [23] K. Ramirez-Amaro, M. Beetz, G. Cheng, Transferring skills to humanoid robots by extracting semantic representations from observations of human activities, *Artificial Intelligence*, 2017, 247, 95–118
 - [24] G. Konidaris, L. P. Kaelbling, T. Lozano-Perez, From skills to symbols: learning symbolic representations for abstract high-level planning, *Journal of Artificial Intelligence Research*, 2018, 61, 215–289
 - [25] D. Chandler, *Semiotics: The Basics*, Routledge, London, United Kingdom, 2007, 2 edition
 - [26] S. Harnad, The symbol grounding problem, *Physica D*, 1990, 42, 335–346
 - [27] T. Frasca, B. Oosterveld, E. Krause, M. Scheutz, One-shot interaction learning from natural language instruction and demonstration, *Advances in Cognitive Systems*, 2018, 6, 1–18
 - [28] V. Paléologue, J. Martin, A. K. Pandey, A. Coninx, M. Chetouani, Semantic-based interaction for teaching robot behavior compositions, In: *Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication* (Lisbon, Portugal, 2017), 50–55
 - [29] J. Arkin, M. R. Walter, A. Boteanu, M. E. Napoli, H. Biggie, H. Kress-Gazit, T. M. Howard, Contextual awareness: Understanding monologic natural language instructions for autonomous robots, In: *Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication* (Lisbon, Portugal, 2017), 502–509
 - [30] Y. K. Thu, T. Ishida, N. Iwahashi, T. Nakamura, T. Nagai, Symbol grounding from natural conversation for human-robot communication, In: *Proceedings of the 5th International Conference on Human Agent Interaction* (Bielefeld, Germany, 2017), 415–419
 - [31] M. Stenmark, *Intuitive Instruction of Industrial Robots*, Ph.D. thesis, Lund University, 2017
 - [32] Z. Pan, J. Polden, N. Larkin, S. V. Duin, J. Norrish, Recent progress on programming methods for industrial robots, In: *Proceedings of the 41st International Symposium on Robotics* (Munich, Germany, 2010), 1–8
 - [33] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, O. Khatib, Progress and prospects of the human-robot collaboration, *Autonomous Robots*, 2018, 42(5), 957–975
 - [34] Z. Materna, M. Kapinus, V. Beran, P. Smrž, P. Zemčík, Interactive spatial augmented reality in collaborative robot programming: user experience evaluation, In: *Proceedings of the 27th IEEE International Symposium on Robot and Human Interactive Communication* (Nanjing, China, 2018), 80–87
 - [35] V. V. Unhelkar, J. A. Shah, Enabling effective information sharing in human-robot teams, In: *Workshop on RSS Pioneers* (Pittsburgh, PA USA, 2018)
 - [36] J. Baraglia, M. Cakmak, Y. Nagai, R. P. Rao, M. Asada, Efficient human-robot collaboration: When should a robot take initiative?, *The International Journal of Robotics Research*, 2017, 36(5–7), 563–579
 - [37] T. Munzer, M. Toussaint, M. Lopes, Efficient behaviour learning in human-robot collaboration, *Autonomous Robots*, 2018, 42(5), 1103–1115
 - [38] K. Darvish, F. Wanderlingh, B. Bruno, E. Simetti, F. Mastrogiovanni, G. Casalino, Flexible human robot cooperation models for assisted shop floor tasks, *Mechatronics*, 2018, 51, 97–114
 - [39] J. T. C. Tan, T. Inamura, Integration of work sequence and embodied interaction for collaborative work based human-robot interaction, In: *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction* (Tokyo, Japan, 2013), 239–240
 - [40] K. Collins, A. J. Palmer, K. Rathmill, The development of a European benchmark for the comparison of assembly robot programming systems, In: K. Rathmill, P. MacConaill, P. O’Leary, J. Browne (Eds.), *Robot Technology and Applications*, Springer, Berlin, Heidelberg, 1985, 187–199