Robotics and Autonomous Systems 99 (2018) 148–165

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Integrating multi-purpose natural language understanding, robot's memory, and symbolic planning for task execution in humanoid robots

Mirko Wächter [a,*], Ekaterina Ovchinnikova [a], Valerij Wittenbeck [a], Peter Kaiser [a], Sandor Szedmak [d], Wail Mustafa [c], Dirk Kraft [c], Norbert Krüger [c], Justus Piater [b], Tamim Asfour [a]

[a] Karlsruhe Institute of Technology (KIT), Adenauerring 2, 76131, Karlsruhe, Germany
[b] University of Innsbruck (UIBK), Technikerstr. 21a, 6020 Innsbruck, Austria
[c] University of Southern Denmark (SDU), Campusvej 55, 5230 Odense M, Denmark
[d] Aalto University, Konemiehentie 2, 6020 Espoo, Finland

ARTICLE INFO

Article history:
Received 9 February 2017
Received in revised form 11 September 2017
Accepted 18 October 2017
Available online 14 November 2017

Keywords:
Structural bootstrapping
Natural language understanding
Planning
Task execution
Object replacement
Humanoid robotics

ABSTRACT

We propose an approach for instructing a robot using natural language to solve complex tasks in a dynamic environment. In this study, we elaborate on a framework that allows a humanoid robot to understand natural language, derive symbolic representations of its sensorimotor experience, generate complex plans according to the current world state, and monitor plan execution. The presented development supports replacing missing objects and suggesting possible object locations. It is a realization of the concept of structural bootstrapping developed in the context of the European project *Xperience*. The framework is implemented within the robot development environment *ArmarX*. We evaluate the framework on the humanoid robot ARMAR-III in the context of two experiments: a demonstration of the real execution of a complex task in the kitchen environment on ARMAR-III and an experiment with untrained users in a simulation environment.

1. Introduction

One of the goals of the humanoid robotics research is to model human-like information processing and the underlying mechanisms for dealing with the real world. This especially concerns the ability to communicate and collaborate with humans, adapt to changing environments, apply available knowledge in previously unseen situations. The concept of *structural bootstrapping* introduced in the context of the *Xperience* project [1] addresses mechanisms of employing semantic and syntactic similarity to infer which entities can replace each other with respect to certain roles. For example, if the robot is asked to bring a lemonade but cannot find it in the kitchen, the robot can suggest to replace it with another beverage, e.g. a juice, based on the similarity of both objects being drinkable. Thus, structural bootstrapping allows the robot to reason about an observed novel entity and its potential functionality and features. Earlier experiments demonstrated how structural bootstrapping can be applied at different levels of a robotic architecture including a sensorimotor level, a symbol-to-signal mediator level, and a planning level [2–4].

Structural bootstrapping is related to the concept of *affordances* — latent "action possibilities" available to an agent towards an object, given agent capabilities and the environment [5]. For example, a bowl can afford pouring into it or stirring in it and a knife can afford cutting with it. Object affordances can also be used to support object categorization and infer potential object replacements. For example, if two containers afford pouring into them, then they are interchangeable towards this action. In the context of structural bootstrapping, the interplay between the symbolic encoding of the sensorimotor information, prior knowledge, planning, plan execution monitoring, and natural language understanding plays a significant role. Natural language (NL) commands and comments can be used to set goals for the robot, update its knowledge, and provide it with feedback. Symbolic representations of object affordances can be used for object replacement. Symbolic representations of robot's observations are required to generate realistic plans. Plan execution monitoring is needed to

* Corresponding author.
E-mail addresses: Waechter@kit.edu (M. Wächter), Ovchinnikova@kit.edu (E. Ovchinnikova), Wittenbeck@kit.edu (V. Wittenbeck), Kaiser@kit.edu (P. Kaiser), Sandor.Szedmak@aalto.fi (S. Szedmak), Mustafa@mmmi.sdu.dk (W. Mustafa), Kraft@mmmi.sdu.dk (D. Kraft), Norbert@mmmi.sdu.dk (N. Krüger), Justus.Piater@uibk.ac.at (J. Piater), Asfour@kit.edu (T. Asfour).

https://doi.org/10.1016/j.robot.2017.10.012
0921-8890/

check if a plan was executed successfully and to solve encountered problems, e.g. replace missing objects. The main aspects and contributions of the manuscript are:

- We elaborate on a framework that allows the robot to understand natural language, generate symbolic representations of its sensorimotor experience, generate complex plans according to the current world state, and monitor plan execution. The framework is implemented within the robot development environment *ArmarX*,[1] see also [6]. The developed natural language understanding (NLU) pipeline is intended for a flexible multi-purpose human–robot communication. Given human utterances, it generates goals for a planner, symbolic descriptions of the world and human actions, and representations of human feedback. It grounds ambiguous NL constructions into the sensorimotor experience of the robot and supports complex linguistic phenomena, such as ambiguity, negation, anaphora, and quantification without requiring training data. The NLU component interacts with related components in a system architecture such as the robot's memory, a planner, a replacement manager.
- We address the mapping of sensorimotor data to symbolic representations required for linking the sensorimotor experience of the robot to NLU and symbolic planning and describe how a symbolic domain description is generated from the robot memory each time an NL utterance needs to be interpreted.
- We introduce the novel Replacement Manager (RM) component of the framework, which is responsible for finding possible locations of missing objects as well as replacing missing objects with suitable alternatives. The RM utilizes a variety of replacement strategies based on robot's previous experience, common-sense knowledge extracted from text corpora, visual object features, and human feedback.

Parts of the cognitive architecture presented in this manuscript rely on our previous work. The concept of structural bootstrapping is introduced in the European project Xperience and in [4]. The *ArmarX* framework is introduced in [6]. Our NLU pipeline and the mapping of sensorimotor data to symbolic representations are first presented in [8]. The common sense knowledge extraction from text is described in [9], while the first experiment on object replacement based on text-derived knowledge is described in [3]. Vision-based object replacement is in focus of [10]. The learning framework for computing probable replacements is outlined in [11,12]. The novelty of this manuscript lies in realization of the concept of structural bootstrapping within a control architecture of a humanoid robot and the demonstration of how this concept allows for grounded cognitive behaviors in a complex setting requiring human–robot communication and collaboration. More specifically, we introduce a novel component of the architecture — the Replacement Manager, which implements both previously developed and novel replacement strategies, see Section 5. We extend existing components of the architecture, such as NLU and plan execution and monitoring, with novel functionality allowing the components to interact with the RM, see Sections 4 and 6. For testing the framework, we design and conduct two novel experiments, see Section 7. We test the developed framework on the humanoid robot ARMAR-III [7] in a scenario requiring planning based on human–robot communication. Two experiments are described. First, we present a demonstration of the scenario execution on ARMAR-III in a kitchen environment. Second, we test how well the framework can be employed by untrained users. To do so, we ask the subjects to control the robot in a visual simulation environment by using natural language.

[1] https://armarx.humanoids.kit.edu.

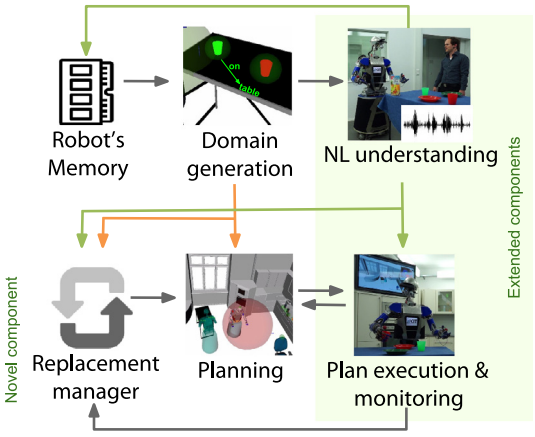

Fig. 1. The system architecture.

The remainder of the article is structured as follows. After presenting the general system architecture in Section 2, we present the domain description generation from the robot's memory (Section 3). Section 4 introduces the natural language understanding pipeline and Section 5 presents the Replacement Manager. Section 6 briefly presents the planner employed in the experiments and discusses how plan execution and monitoring are organized in our framework. Experiments on the humanoid robot ARMAR-III and in a visual simulation environment are presented in Section 7. Related work is discussed in Section 8, while Section 9 concludes the manuscript.

2. System architecture

The system architecture is implemented within the robot development environment *ArmarX* [6] and is shown in Fig. 1. The system architecture consists of six major building blocks: robot's memory, domain generation, natural language understanding, replacement manager, planning, as well as plan execution and monitoring.

The robot's memory is represented within *MemoryX*, one of the main modules of the *ArmarX* architecture (see Section 3). The memory stores and offers symbolic and sub-symbolic information about prior knowledge, long-term knowledge and knowledge about the current world state. Domain descriptions in symbolic form are generated from the robot's memory (see Section 3). The domain descriptions are also used by the NL understanding component for grounding and generating the domain knowledge base, see Section 4. The developed multi-purpose NLU framework can distinguish between (a) direct commands that can be executed without planning (*Go to the table*), (b) plans requiring commands that are converted into planner goals, which are processed by a planner (*Set the table*), (c) descriptions of the world that are added to the robot's memory and used by the planner (*The cup is on the table*), and (d) human feedback (*I'm fine with it*). The goal generated by the NLU component is passed to the Replacement Manager that checks for each object in the goal if this object and its location are in the domain description (Section 5). If an object or its location are unknown, then the component suggests available alternatives. NLU also provides the Replacement Manager with context-based affordances for mentioned objects. After suggesting a replacement, the Replacement Manager waits for a human confirmation or rejection, which is provided by the NLU component. If a replacement is confirmed, then the goal is rewritten and passed to the planner together with the domain description. The planner takes a domain