Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-21 (2016) 668–674

Human robot cooperation planner using plans embedded in objects ⋆

Hatem Fakhruldeen * **Pranav Maheshwari** **
Alexander Lenz *** **Farid Dailami** **** **Anthony G. Pipe** †

* *Bristol Robotics Laboratory, Bristol, United Kingdom (e-mail: hf13415@bristol.ac.uk).*
** *Bristol Robotics Laboratory, Bristol, United Kingdom (e-mail: pmaheshw@andrew.cmu.edu).*
*** *Bristol Robotics Laboratory, Bristol, United Kingdom (e-mail:alex.lenz@brl.ac.uk)*
**** *Bristol Robotics Laboratory, Bristol, United Kingdom (e-mail:farid.dailami@uwe.ac.uk)*
† *Bristol Robotics Laboratory, Bristol, United Kingdom (e-mail: tony.pipe@brl.ac.uk)*

Abstract: A human robot cooperation (HRC) planner for joint assembly tasks is presented. By combining first order logic (FoL) planning and object oriented programming, plans embedded in objects (PeO) can be produced. This concept shifts the focus of planning from actions to objects, thus defining them in terms of objects and combinations of their actions. The produced plans are represented using tree structures, which capture the task execution status in terms of completeness, agents' roles and actions' order. The planner and its application program interface (API) were written and developed using SWI-Prolog. This API allowed the planner to take various inputs to create different plans, and simultaneously to get a variety of useful outputs. The planner was tested in a collaborative pipe assembly task as a part of a bigger architecture implemented on the Baxter robot. It was able to generate the plan and achieve successful execution. This proof-of-concept planner shows great potential; allowing for a novel description in the HRC domain and execution of human-robot interaction tasks with industrial production in mind. However, first results are presented here; more work is required to take the work to higher technology readiness levels.

Keywords: Robotics, Human robot interaction, Planning, Artificial intelligence, Assembly robots

1. INTRODUCTION

Collaboration is one of the hallmarks of human culture. Humans cooperate together to achieve feats that are unachievable by single individuals (e.g. group hunting, bridge building etc). Furthermore, human children learn this skill from their early days, as they can start cooperating with other children or older individuals in an effortless manner in cooperative games and tasks. It is understood that this skill, possessed in this highly developed form, is unique to humans among primates (Dominey and Warneken, 2011). Robots engaged in Human Robot Interaction (HRI) are still rather considered as tools than partners. Many safety measures are taken to protect humans and the level of interaction is still very limited to very few scenarios. Overall, comparing Human-human interaction and HRI shows that HRI is well below par.

However, lots of research has been done to close the gap between Human-Human Interaction and HRI. Especially in the area of Human robot collaboration (HRC), which involves shared plans between two agents. The works by Jevtic et al. (2012), Lallee et al. (2012), Mainprice and Berenson (2013), Breazeal et al. (2004) and Lallee et al. (2009) made great progress in raising the robot capabilities and making them more efficient and effective in those collaborative tasks.

A major focus of HRC research is planning in terms of requirements, representation and plan production in cooperative tasks. Humans' ability to have a "bird's eye view" of the shared plan and clear understanding of their roles, were found to be required in order to achieve successful collaboration (Lallee et al., 2009). Furthermore, using comprehensive and compact representation for plans is another important point. This representation has to be suitable for the context of its use. The work of Lenz et al. (2012) used such a representation, where a hierarchical assembly plan was constructed using a collection of atomic actions and their combinations. As for the production of such cooperative plans, a number of planners took the human intentions and actions into account and produced plans that were proactive in nature such as in the works

⋆ This work was undertaken during the EPSRC (via the Centre in Innovative Manufacturing in Intelligent Automation) funded feasibility study named 'HRI Team Work Study' (HRITWS) at Bristol Robotics Laboratory

10.1016/j.ifacol.2016.10.677

of Schrempf et al. (2005), Talamadupula et al. (2014) and Alami et al. (2005). Other planners used insights from successful human teams such as planning on the fly, communicating progress and sharing mental models between all the agents to achieve efficient collaboration (Shah et al., 2011).

However, In the works of Jevtic et al. (2012), Lallee et al. (2012), Breazeal et al. (2004), Lallee et al. (2009), Lenz et al. (2012), Schrempf et al. (2005) and Alami et al. (2005), plans, which were composed from sequences of actions or skills, were either learned or preprogrammed by the human users. In addition, the order and roles of actions were assigned pre execution between both actors (human and robot). In our work here, we are introducing a simple HRC shared planner that generates plans from the task's objects and their relationships, rather than learned or programmed actions sequences, thus shifting the planning focus from the available actions to the available objects. These plans are suitable for assembly tasks, where they are produced by calling the final assembled object. Furthermore, they allow changing the actions' order and their roles online. Hierarchical tree structures are used to represent these plans, which are compact and insure bird's eye view of the whole plan, thus easy understanding of the agents roles and responsibilities. This approach is similar to the concepts arising from considerations of cyber-physical systems and the 'Internet of Things', typified by Industry 4.0 (Lee, 2008). A planning problem is formulated and solved using Prolog programming language, FoL and object oriented programming concepts. This planner shares some concepts with classic planners such as STRIPS (Fikes et al., 1998).

In the reminder of this paper, we explain in Section 2 the concept behind PeO and how plans relate to objects. Section 3 describes our implementation of PeO in SWI-Prolog (Wielemaker et al., 2012) and the developed API to interact with this planner. In Section 4 we present an application of this concept in a pipe assembly cooperative task. Finally, Section 5 concludes this approach and provides an outlook for future works to expand our concept.

2. CONCEPT DESCRIPTION

2.1 Objects

The basic idea of PeO is that each object has a set of plans that are defined in terms of actions. Those actions are related to the object itself or other objects. For instance in an assembly setting, the object *nut* has two actions which are: 'fasten a bolt' and 'be manipulated in the workspace'. Those actions are related to this object alone. Similarly, the object *bolt* has two actions: 'be fastened by a nut' and 'be manipulated in the workspace'. The combination of those items and their actions, results in plans that produce assembled objects, such as an *assembled bolt and nut*. Those produced objects also contain their own actions, which when combined with other objects result in more complex plans.

After this short description, we can start to describe our concept in more depth. Objects in PeO are categorized to two classes.

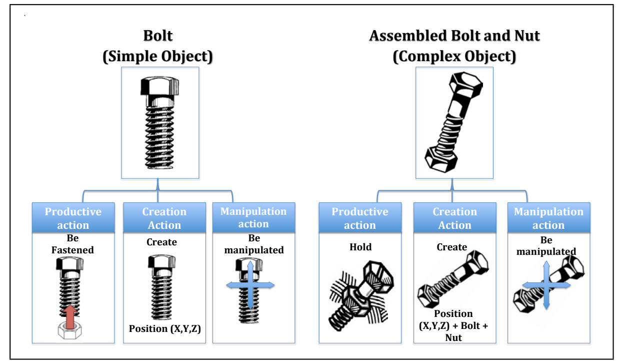

Fig. 1. the Different components of PeO explained in the context of a bolt and nut example.

(1) *Basic objects*: are not the product of any assembly. They are assumed to be existent in the world and don't need specific plans to be assembled. i.e. a nut which can't be created from an assembly. Their creation actions only bring them into known existence.
(2) *Complex objects*: are the product of assembly of two or more basic objects, e.g. an assembled bolt and nut. Their creation actions involve an assembly of objects.

Similarly, actions which an object can possess are divided into three types:

(1) *Manipulation actions*: involve translating and rotating the object around in the world.
(2) *Creation actions*: bring objects into existence. Those actions are the building blocks of our plans. For basic objects those actions simply involve defining the initial position of the object in the world. For complex objects, the creation actions involve a set of objects (basic and complex) along with their productive actions that need to be performed in order to create the required complex object.
(3) *Productive actions*: involve actions such as mounting, drilling, fastening, spraying, etc. They can be performed on other objects, or on the object itself. They result in changing the state of the world, by altering the state of objects, or producing new complex objects when combinations of them occur.

2.2 Plans

We will use the example shown in Fig. 1 to illustrate this concept. In order to create the bolt and nut assembly, a nut would need to be fastened to a bolt. So, the plan's basic objects are *bolt* and *nut*, while its final goal is the production of the complex object *assembled bolt and nut*. The object *bolt* has the manipulation action (*be_manipulated*), which can move it around the work space. Its creation action (*create*) that brings the bolt into existence in the work space. It simply defines its initial position and makes it available for any other actions. Finally, there is its productive action (*be_fastened*), which when combined with the *nut* action (*fasten*) produces the *assembled bolt and nut*.

The *nut* manipulation and creation actions are the same as the bolt, but its productive action is different. Even though both actions perform the act of 'fastening', in one of them the object is the grammatical subject of the action