Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling

# ROSPlan: Planning in the Robot Operating System

**Michael Cashmore, Maria Fox, Derek Long, Daniele Magazzeni, and Bram Ridder**
King's College London
London WC2R 2LS
*firstname.lastname*@kcl.ac.uk

**Arnau Carrera[a], Narcís Palomeras[b], Natàlia Hurtós[b], and Marc Carreras[a]**
University of Girona
17071 Girona, Spain
[a]*firstname.lastname*@udg.edu
[b]n*lastname*@eia.udg.edu



**Abstract**

The Robot Operating System (ROS) is a set of software libraries and tools used to build robotic systems. ROS is known for a distributed and modular design. Given a model of the environment, task planning is concerned with the assembly of actions into a structure that is predicted to achieve goals. This can be done in a way that minimises costs, such as time or energy. Task planning is vital in directing the actions of a robotic agent in domains where a causal chain could lock the agent into a dead-end state. Moreover, planning can be used in less constrained domains to provide more intelligent behaviour. This paper describes the ROSPLAN framework, an architecture for embedding task planning into ROS systems. We provide a description of the architecture and a case study in autonomous robotics. Our case study involves autonomous underwater vehicles in scenarios that demonstrate the flexibility and robustness of our approach.

## 1 Introduction

Planning is concerned with organising instances of actions in order to achieve certain goals (Ghallab, Nau, and Traverso 2004). It begins with a domain model describing the actions available to the planner and a description of the current state. The actions are then assembled into a structure that is causally valid, with an attempt to optimise some cost function. In order to do this, planning must forecast interactions with future constraints, avoid moving the executor into dead-end situations, and still achieve the goals.

The Robot Operating System (ROS) (Quigley et al. 2009) is a set of software libraries and tools used in building robotic systems. ROS has become a popular platform for robotics research and has also proved a flexible foundation on which to build robotic control via task planning (Bernardini, Fox, and Long 2014; Cashmore et al. 2014; Dornhege, Hertle, and Nebel 2013).

Combining task planning and robotics presents several challenges, in particular:

- Given a domain model that matches the capabilities of the robot, an initial state must be generated that matches the current environment.
- Actions planned by the task planner, on an abstracted model of the world, must be made concrete and dispatched to lower level controllers.
- Plans must be executed according to some strategy. This must account for action failure, plan failure due to ignorance or change in a dynamic environment, and changing mission requirements.

We introduce ROSPLAN[1]. ROSPLAN is a framework for embedding a generic task planner in a ROS system. The architecture is highly modular and deals with the stated challenges by providing tools to: automatically generate the initial state from the knowledge parsed from sensor data and stored in a knowledge base; automate calls to the planner, then post-process and validate the plan; handle the main dispatch loop, taking into account changing environment and action failure; and match planned actions to ROS action messages for lower level controllers.

Integrating planning with robotics through a plan execution architecture has been done successfully by others, for example by McGann et al. (2008) in the T-REX system. T-REX is a timeline based plan execution architecture supporting distributed deliberation amongst a collection of *reactors*. Each physical component of the robotic system comprises one or more inter-dependent reactors which are responsible for evolving state variables on the different timelines. Each reactor has a look-ahead, determining how far ahead it can plan, and a latency, bounding the planning time available to it. The system is synchronised using a clock, and concurrent activity is achieved by enforcing inter-dependencies between the timelines of the relevant state variables. Generative planning is done using the Europa system (Frank and Jonsson 2003) with a timeline-based modelling language. T-REX has been used successfully to plan and execute underwater AUV missions at the Monterey Bay Aquarium Research Institute (MBARI) (Graham et al. 2012; Py, Rajan, and McGann 2010; Magazzeni et al. 2014). Task planning has also been embedded into robotic systems in a number of other ways. In particular, Ponzoni et al. (2014) describe a configurable anytime meta-planner that drives a (PO)MDP planner, anticipating the probabilistic evolution of the system; Srivastava et al. (2014) provide



[1]The source code and documentation for ROSPlan can be obtained from *https://github.com/KCL-Planning/ROSPlan*