# ROS-TiPlEx: A collaborative design tool for Timeline-based Planning & Scheduling applications with ROS

Carlo La Viola, Andrea Orlandini, Alessandro Umbrico, and Amedeo Cesta

Institute of Cognitive Science and Technology, National Research Council of Italy
Via San Martino della battaglia 44, 00185 Rome, Italy
name.surname@istc.cnr.it

**Abstract.** This paper provides a quick overview of a Knowledge Engineering system, called ROS-TiPlEx (*Timeline-based Planning and Execution with ROS*), to provide a shared environment in which experts in robotics and planning can easily interact to, respectively, encode information about low-level robot control and define task planning and execution models. ROS-TiPlEx aims at facilitating the interaction between both kind of experts, thus, enhancing and possibly speeding up the process of an integrated control design. ROS-TiPlEx is the first tool addressing the connection of ROS and timeline-based planning.

## 1 Introduction

Deploying interactive robots in human-populated scenarios requires to address multiple challenges. Among others, it is of paramount importance the ability of the robot to quickly adapt its behaviours to the actual state of the environment and to keep the user engaged in the interaction. Such highly flexible and adaptive behaviours are necessary also to guarantee safe and effective human-robot interactions. There are several approaches that aim to achieve robust action selection via planning, e.g., [2, 3, 7] or robust execution via some form of finite state machine (FSM), e.g., [1, 12]. The introduction of tools to facilitate the integration of planning and robotics and the design of well-integrated solutions entails different kind of expertise to address a wide variety of control issues, spanning from low-level control to decisional autonomy configuration. The variety of the topic introduces some engineering problems; information sharing at different abstraction levels may cause redundancies or inconsistencies in planning specifications; the lack of a generally accepted design methods may entail many potential back-and-forth (re)work over models and control parameters before defining the proper control configuration. A workaround for this solution is to use state-of-the-art software tools which are in most of the cases developed to support either robotics or planning experts [2] and not the overall process as a collaborative work. In this paper we present a tool that allows to overcome these problems, acting as a base for information sharing and monitoring of the overall process.

## 2 A tool to foster collaboration

In real-world application the design of a Human-Robot Interaction (HRI) system, presents various needs that can not be fulfilled by a single person. The main issue is the variety of information that must be processed all together to reach the final result. Often the contribute of the expert is specific to its field and not integrated with the rest of the solutions; each actor takes in charge single *building-blocks* of the process to which he contributes with his expertise. The main role in the construction of such scenario is focused on the *mission expert*, or rather the person that knows the mission objectives, and is able to dispatch the work to the engineers that best match the competencies for the specific task. Such an expert will be responsible for the correct execution of the separated tasks and it will be the main point of contact in case of any issue in the project. But if the *mission leader* is responsible for the overall success of the project, then other figures will have responsibility on the single building-blocks of the system. For the sake of brevity, we can summarize the main figure of expertise in a *planning expert* and a *robot expert*. The first has knowledge of P&S, and can provide robust planning for the system, to reach the standard of quality and safety that must be met for any HRI application; the second, on the other hand, is the one responsible for the robot itself. He should provide hardware capable of satisfying the mission objective and equipped with all the tool that is needed by the specific application (e.g. robot arms, camera, working tools, etc...). If we want to summarize the main roles and tasks:

1. *Mission leader*: Responsible for the whole project and in charge of dispatching the tasks to the people with the matching competencies;
2. *Planning expert*: In charge of creating a robust P&S domain to reach the standard of quality and safety imposed by the mission leader; also responsible for the validation and real-time execution of the plan;
3. *Robot expert*: In charge of the hardware platform that must provide all the tool needed to accomplish the mission objectives defined by the mission leader. It is also responsible for the quality of the hardware.

Such different roles will most likely have different background and different vision of the system. Not to mention other constraints like geographic position (the experts could be in different parts of the globe). If we consider the heterogeneity of such scenario, it is clear the need for a shared framework that helps the different actors to collaborate, share the same vision, and have a common platform where they can communicate. ROS-TiPlEx aims to fill the gap and promotes communication among these different figures, even though in its first implementation it can't support all the actors in the integrity of the process, even though automates the most part of it.

## 3 ROS-TiPlEx

**The context.** ROS-TiPlEx([6]) has been developed with the main focus on re-usability and capability to adapt with minimum effort to different scenarios.

Apart from the differences among all the actors, the robotic world also presents a multitude of technologies [9], and there is need to develop a tool that is built on shared standard, in order to promote reuse and information sharing. With this objective, ROS-TiPlEx is built on top of ROS, a standard in robotics that has already shown the capacity of integration with planning software [2]. ROS is an operating system for robots and offers an abstraction of the physical layer, offering libraries that simplify the control of the low-level components. The other main technological pillar of ROS-TiPlEx is timeline-based P&S [8], which is integrated through the use of a framework, called PLATINUm[11, 10]. It is worth underscoring that ROS-TiPlEx is built on top of a formal framework of Timeline-based P&S [4], hence it can be adapted to work with any planning software that exploits such a paradigm.

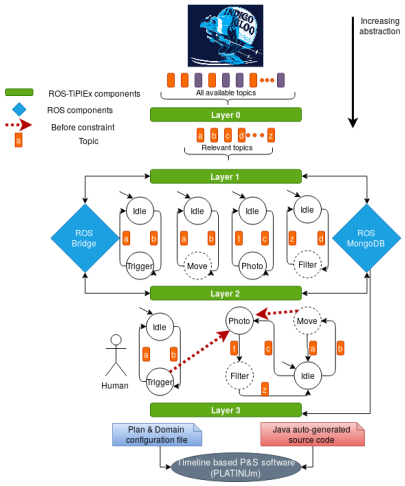


**Fig. 1.** ROS-TiPlExarchitecture overview

**The architecture.** ROS-TiPlEx is designed with a Layered architecture, that allows introducing progressive levels of abstraction throughout the process, which lead to the modularity of the whole process. The process starts from *Layer 0*, that is in contact with the ROS platform (i.e., the robotic platform), and is able to access its data. In the specificity of ROS, layer 0 behaves as a publisher and subscriber that registers to the ROS topics involved in the execution of the mission tasks. Once the system is scanned, the process can pass to *Layer 1*, where the robot expert should start his activity. Knowing the robot and the premises given by the *mission leader*, he should firstly define an atomic action of the robot (e.g. move from A to B), and map such action on an FSM (Finite State Machine), where the edges correspond to the topics involved in the action, and the states are unique names decided by the expert in relation to the specific

domain or mission goals. In this FSM, when a message passes on the topic related to the edges, a change of state is triggered and reflected on the P&S system. The manipulations on this layer are allowed thanks to two components integrated into ROS-TiPlEx: the *ROS-bridge component* that allows the access to the system thanks to a Javascript UI, and the *Mongo-DB component*, that stores the data in the DB and allows them to be reused. These same components are the one used on *Layer 2*, granting now the access for a planning expert. He has access to the atomic actions created in the previous layer and can manipulate them, to transform the simple FSMs in more complex ones. Such conversion is still based on the specification given by the mission leader and is needed to prepare a suitable model for the P&S domain, that would allow improved performances and stability when it comes to elaborate the planning domain and the execution plan. After the rework performed by the planning expert, the information in the system goes through the last block, the *Layer 3*, whose purpose is to convert all the data in files that will serve as input for the P&S software. In the case of PLATINUm, the output will be composed by the domain description and some Java classes which still need to be updated, in order to allow the execution of the plan that will be dispatched by PLATINUm.

## 4 Use case

A simple yet significant use case has been deployed together with the ROS-TiPlEx, to test and evaluate the software. It is possible to run the software using the code in [5], that provides all the information needed for the execution. In our usage scenario, a mission leader defines the mission goals that can be summarized in 1) Autonomous movement in unknown space; 2) Ability to take photos of specific points of interest, when a human operator triggers the action through a remote controller; 3) Apply analytics and extract information from the pictures. At the first stage of the process, the first actor to be involved in the configuration is the robot expert. He is supposed to know the robot and its capabilities, hence know how to correctly trigger the robot to reach the mission objectives. Through the use of ROS-TiPlEx, he can input such information and elaborate the FSMs in figure 2. As we can notice, such FSMs are very basic and unlikely to bring a

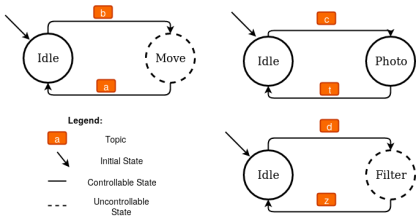


**Fig. 2.** FSMs elaborated by robot expert for the robot actions

real added value when it comes to the generation of a P&S domain; it is for this reason that a further refinement is needed by the planning expert, that can use his understanding of the mission objectives to generate State Variables, that will provide a suitable input for the P&S software, PLATINUm in the specific case. To construct a state variable, the planning expert accesses ROS-TiPlEx on the dedicated interface and can read the inputs from the robot expert that where previously stored in the MongoDB ROS node. When reworking such information, it is possible to introduce new constraints among the states, merge the FSMs to obtain a more structured datum, and also introduce *Temporal Constraints*, that will force the system to behave as expected by the mission leader. In the

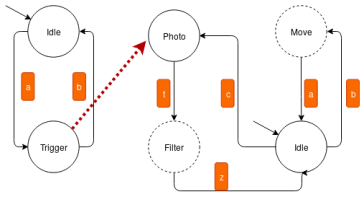


**Fig. 3.** State Variable after rework of the planning expert

example of figure 3, the *Photo* state must always be preceded by a transition of the first State Variable through the *Trigger* state. Such condition implements the rule for which it is the human inside the system that must interact with the robot and send commands to it at the right time.

## 5 Conclusions and further improvements

In this first implementation of ROS-TiPlEx, we have shown how it is possible to build an engineering tool that allows the communication and the information sharing of different entities, with different skills, such as a robotic expert and a P&S expert. Albeit this strategy has been presented and experimented, there still are improvements that will lead to a more complete and more usable tool. A first advancement should be done on adding to ROS-TiPlEx a module that allows the *mission leader* to define an unambiguous description of the mission scenario so that the experts participating to the process can have a common vision of the final result. Nonetheless, the shared point of information could be accessible from different geographical points, enabling the faculty to work on the same topic even without a physical presence. Another advancement needed in ROS-TiPlEx is to improve the UI implementation, that now is not very satisfactory when it comes to user experience or usability. Moreover, an overall refactoring of the code is needed to improve the separation of the front and the back end, enabling better portability and the faculty to split the components in multiple machines,

centralizing on the robot only what is strictly needed to the elaboration and execution of the plan. Last but not least, ROS-TiPlEx could be equipped with one level more of abstraction, in order to make its implementation independent from the actual ROS version, because this OS is in constant evolution and a new tool can not be limited by the versioning of the underlying platform. In conclusion, the added value of ROS-TiPlEx is to bring a shared vision in the process of development related to the P&S applied to robotics, on the for of an engineering tool that is able to connect different experts with different needs, and help them to reach the common goal of the the mission scenario satisfaction.

## References


1. Bohren, J., Cousins, S.: The smach high-level executive [ros news]. IEEE Robotics Automation Magazine **17**(4), 18–20 (2010)
2. Cashmore, M., Fox, M., Long, D., Magazzeni, D., Ridder, B., Carrera, A., Palomeras, N., Hurtos, N., Carreras, M.: Rosplan: Planning in the robot operating system. In: Proceedings of ICAPS. vol. 2015-January, pp. 333–341. AAAI press (2015)
3. Chanel, C.P.C., Lesire, C., Teichteil-Knigsbuch, F.: A robotic execution framework for online probabilistic (re)planning. In: Proceedings of ICAPS. AAAI press (2014)
4. Cialdea Mayer, M., Orlandini, A., Umbrico, A.: Planning and execution with flexible timelines: a formal account. Acta Inf. **53**(6-8), 649–680 (2016)
5. La Viola, C.: Ros-tiplex docker and installation details. https://github.com/carloLV/ros-tiplex-docker, accessed: 2019-07-22
6. La Viola, C., Orlandini, A., Umbrico, A., Cesta, A.: Ros-tiplex: How to make experts in a.i. planning and robotics talk together and be happy. In: 28th IEEE International Conference on Robot & Human Interactive Communication (ROMAN 2019) (2019)
7. Lacerda, B., Parker, D., Hawes, N.: Optimal and dynamic planning for markov decision processes with co-safe ltl specifications. In: IEEE International Conference on Intelligent Robots and Systems. pp. 1511–1516. Institute of Electrical and Electronics Engineers Inc. (2014)
8. Muscettola, N.: HSTS: Integrating Planning and Scheduling. In: Zweben, M. and Fox, M.S. (ed.) Intelligent Scheduling. Morgan Kauffmann (1994)
9. Quigley, M., Gerkey, B.P., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: Ros : an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
10. Umbrico, A., Cesta, A., Cialdea Mayer, M., Orlandini, A.: Integrating resource management and timeline-based planning. In: The 28th International Conference on Automated Planning and Scheduling (ICAPS) (2018)
11. Umbrico, A., Cesta, A., Cialdea Mayer, M., Orlandini, A.: Platinum: A new framework for planning and acting. In: AI*IA 2017 Advances in Artificial Intelligence. pp. 498–512 (2017)
12. Ziparo, V.A., Iocchi, L., Lima, P.U., Nardi, D., Palamara, P.F.: Petri net plans. Autonomous Agents and Multi-Agent Systems **23**(3), 344–383 (2011)