Coloured Petri Net Plans for cooperative multi-robot systems

Lorenzo Steccanella[b], Alessandro Farinelli[a], Luca Iocchi[b] and Daniele Nardi[b]

a: Dipartimento di Informatica, Università degli studi di Verona
b: Dipartimento di Ingegneria informatica automatica e gestionale,
Sapienza Università di Roma

1 Introduction

During last years the field of Multi-Robot Systems (MRS) has developed significantly growing in size and importance. There exist numerous areas where multi-robot systems have been used successfully and, in the majority of them, MRS must execute complex tasks in environments that are dynamic and unpredictable. This has led to the problem of synthesis and monitoring of complex plans that can provide high level commands to the system allowing the specification of parallel actions, interruptions of task in execution, synchronization between robots, and so forth.
Petri Nets (PN) [1,2] have recently emerged as a promising approach for modeling either single-robot or multi-robot plans. This approach provides a clear graphical representation for modeling and developing systems which are concurrent, distributed, asynchronous, non-deterministic and/or stochastic.
One of the issues of the approaches that uses Petri Nets is the space complexity associated to the specification of the plans, which can become very large (i.e., with many graphical elements), especially in the case of multi-robot systems.

In this work, we analyse the use of Coloured Petri Net (CPN) [3] for the creation and validation of multi-robot systems. More specifically, we describe a formalism for representing multi-robot plan by using CPN and an algorithm to translate the CPN plan in a Petri Net Plan (PNP) [4].
PNP is a plan specification language based on Petri Nets that has been widely used for several robotics applications ranging from robotic soccer to search and rescue and service robotics[1]. PNP are based on PNs and the support for multi-robot plans is obtained by specifying the name of the robot or of the role within the description of each action. This features allows for easy implementation of centralized and distributed plans, but it is suitable for situations where the number of robots/roles is limited.

CPNs differ from PNs in one significant respect; tokens can be of different types which are usually called *colours*. Hence, places in CPN can contain a multi-set of coloured tokens and the firing rules associated to transitions depend on such colours. As a consequence, Coloured Petri Nets are equivalent to Petri Nets with respect to descriptive power [7] but provide a more compact plan specification and are particularly well suited for multi-robot plans [6]. The use of CPN for modelling multi-robot plans has the advantage of using coloured tokens to represent different robots/roles and thus of improving its scalability.

[1] pnp.dis.uniroma1.it

2 Methodology

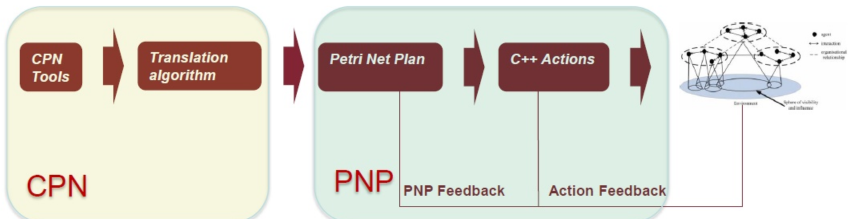

Fig. 1: System architecture

The general idea behind our methodology is described in Figure 1. We start by designing the multi-robot plan by using Coloured Petri Nets, we then translate the Coloured Petri Net Plan in a Petri Net Plan and we execute the plan within the Petri Net Framework [4]

Our work is based on two main ingredients: first, we elaborate a specific CPN formalism that can be translated to executable plans for the PNP framework. Second, we provide an algorithm for the automatic conversion of a such CPN plans in PNP plans. Specifically, in our CPN plan we have three basic elements that compose a multi-robot plan:
i) Action: An action represents an operation that the robots should execute in the world (e.g., explore a given region of the environment). Following the PNP framework an action can be preceded by a start transition, that defines the preconditions for executing the action. We then have an execution place that represents the fact that the action is being executed and finally an end transition that follows the execution place and fires whenever the end conditions for the actions are true (see Figure 2). However, in contrast to PNP the execution place associated to an action can contain a multi-set of coloured tokens, indicating that a set of robots are executing the action in parallel.
ii) Operator: An operator is a place used for the control of the behavior of the net, that is not related to any action in the real word. For example, Figure 3 shows the use of an operator to connect the end and start transitions of Action 1 and Action 2 respectively. Notice that, the translation of the operator in the PNP framework defines different behaviors and depends on modifiers that we can define on the transitions. For example, the $SYNC$ modifier (specified in the end transition associated to Action 1) results in a PNP plan that synchronizes the parallel actions of all robots (i.e., no robot will start Action 2 until all robots have completed Action 1). Our formalism allows to handle in a similar way other important constructs such as team level or individual level interrupts, i.e., interrupts that change the behaviors of the whole team or of a single robot respectively.
iii) Resources: A resource is a coloured token that can represent a robot or a set of robots. This element determines which robots should execute the actions.

The combination of the above elements allows to design complex multi-robot plan. In particular, thanks to the use of coloured token, we can achieve a signif-

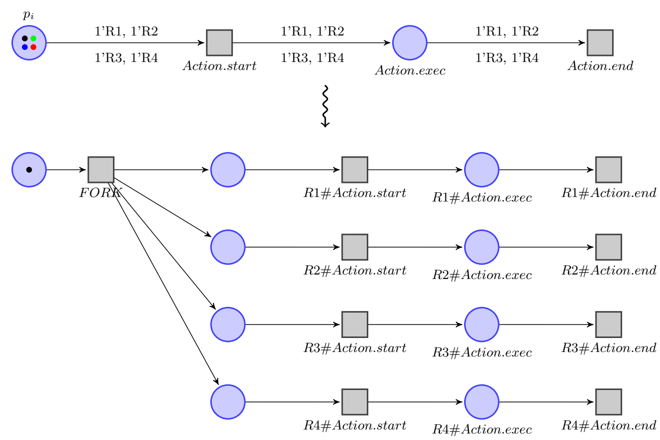

Fig. 2: Conversion of multi-robot Action from CPN to PNP

icant reduction in terms of space, essentially because we can represent parallel actions executed by n robots with a single place that contains n tokens. This is crucial as it significantly eases the design and monitoring tasks for the human operators. Moreover, notice that we can maintain the identity of the robots executing the action: for example, an explore action (represented with a single place in the CPN) could be executed in different ways depending on whether the different coloured tokens represent UGVs or AGVs.

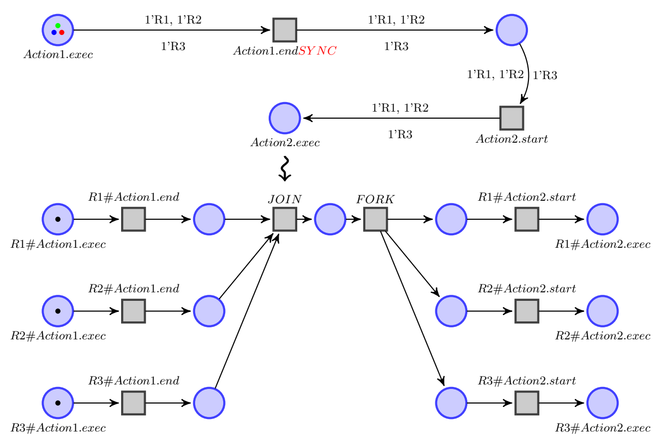

Fig. 3: Conversion of synchronized actions from CPN to PNP

There are several free available softwares that allows the design and the analysis of Coloured Petri Net, here we used the "CPN tools" [5]. Thanks to this software we can design CPN plan in an easy and quick way, and allows to save them in the PNML format, a special well structured XML specific for Petri nets.

A key point of this work is that the compilation of a CPN plan to an executable PNP plan is automatic, and we provide an algorithm (not reported here in the interest of space) to realize such conversion.

3 Results

The proposed system has been evaluated using the Stageros simulator. In particular, we consider three different use cases related to the collaborative exploration of an indoor area. The evaluation confirms the efficacy of the system in terms of space reduction and ease in the design of the plan.
A video of the execution of one of these use cases is available at:`https://youtu.be/9e9x87FCjiY`. This video is a proof of concept for a cooperative exploration scenario, where a team of two heterogeneous robots should explore a pre-specified area. We have two teams, composed by two green and two blue robots respectively, and we want only one team (i.e., the closest one) to perform the exploration. The first part of the video shows the translation of the CPN to PNP, notice that each robot has been represented as a single token so to control each one of them (e.g., for possible single robot interrupts). We have a decision making point at the beginning of the mission, where the plan evaluates the distance of the two team members from the target area (the room in the top left) so to choose which team will perform the explore action (i.e., the team that minimizes the sum of travel distance). In our case this is the blue team. During the exploration, one of the two robots detects a low battery state, hence the current mission is interrupted and this robot goes to a recharge base (red area). Then both robots move to the home position, and since the original specification of the CPN was to synchronize the end of the mission, only when both robots are back to the home position the mission is over (i.e., tokens enter the goal state).

4 Conclusions

In this work we propose a compact and effective representation of multi-robot plans by using Coloured Petri Nets; such a representation is comprehensive, easily readable by humans, has minimum space requirements, and is suitable for real time execution. Experimental evaluation of the considered use cases suggests that our approach is indeed capable of automatically generating executable PNP plans from a CPN description, handling interesting cases such as for example parallel action execution, as well as team level and individual interrupts.
Our work opens several scenarios for future improvements. Specifically, an interesting direction would be to work towards an on-line translator for multi-robot CPN plans hence converting parts of the CPN as the execution progress so to avoid a static conversion of the whole plan. Moreover, while in our work the evaluation is based on a standard and widely used robotic simulation environment, an empirical evaluation with robotic platforms is important to precisely assess the impact of our methodology.

References

1. Murata, Tadao. "Petri nets: Properties, analysis and applications." Proceedings of the IEEE 77.4 (1989): 541-580.
2. C. A. Petri. Communication with automata. PhD thesis, Rome Air Development Center,Rome, NY, 1966.
3. Jensen, Kurt. Coloured Petri nets: basic concepts, analysis methods and practical use. Vol. 1. Springer Science & Business Media, 2013.
4. Ziparo, Vittorio Amos, and Luca Iocchi. "Petri net plans." Proceedings of Fourth International Workshop on Modelling of Objects, Components, and Agents (MOCA). 2006.
5. Anne Vinter Ratzer, Lisa Wells, Henry Michael Lassen, Mads Laursen, Jacob Frank Qvortrup, Martin Stig Stissing, Michael Westergaard, Sren Christensen, and Kurt Jensen. 2003. CPN tools for editing, simulating, and analysing coloured Petri nets. In Proceedings of the 24th international conference on Applications and theory of Petri nets (ICATPN'03), Wil M. P. Van Der Aalst and Eike Best (Eds.). Springer-Verlag, Berlin, Heidelberg, 450-462.
6. Marciano, Limor. CPNP: Colored Petri Net Represention of Single-Robot and Multi-Robot Plans. Diss. Department of Computer Science, Bar-Ilan University Ramat Gan, Israel, 2013.
7. Jensen, Kurt. "A method to compare the descriptive power of different types of Petrinets." International Symposium on Mathematical Foundations of Computer Science. Springer Berlin Heidelberg, 1980.