Plan Monitoring and Validation for Cooperative Robotic Watercraft

Nicolo Marchi*, Masoume M. Raeissi*, Alessandro Farinelli*, Nathan Brooks†, Paul Scerri†

* Computer Science Department,
University of Verona, Italy
†The Robotics Institute,
Carnegie Mellon University, Pittsburgh, U.S.A.

In recent years, Multi-Robot Systems have been frequently used to provide situation awareness to human operators in various contexts, ranging from search and rescue to environmental monitoring [1, 2]. In particular, significant efforts have been recently devoted to robotic systems that can provide information on water quality for lake, rivers and the sea, both above and below the water surface. The use of such autonomous robotic systems has the unique advantage of providing samples for large body of water over time and to correlate such sample with geographical information, hence providing a map of the phenomenon (e.g., Ph, temperature, dissolved oxygen, chemical diffusion) rather than point samples.

In this work we focus on the cooperative robotic watercraft system developed by the CMU Robotics Institute [2]. A crucial feature of such system is the development of low cost platforms (which uses standard devices such as smart-phones for connectivity and localization) and the focus on the cooperative aspects.

A crucial element for the CRW system we consider here is the interaction between the human operators and the boats. The system is designed to have one (or few) human operator to control a set of boats, hence the idea is to provide the platform with autonomous behaviors, such as navigation and exploration, and provide high level commands, such as explore this portion of area or come back home. To realise this a crucial element of the system is the specification of such behaviors through plans specified by using a formalism based on Petri Nets. While Petri Nets has been successfully used before to represent Multi-Agent Plans [3], the system used for the CRW departs from such previous work as it uses Coloured Petri Nets, where each token type specifies a different boat. This results in a compact and effective representation of complex team-level plans, For example, Figure 1 reports a Petri Net that represents an exploration behavior for a set of boats. Notice that, an interesting element for such plan is the possibility to represent user input/interactions, such as the selection of the area to explore (SelectArea transition) or the selection of which boats should participate in the exploration (SelectBoat transition).

However, since the interaction with human operators is crucial for the success of the mission, a desirable feature of the system would be to provide a more sophisticated interaction with the plan execution phase. Specifically, for a plan such as the one reported in Figure 1 there is no way for the human operator to

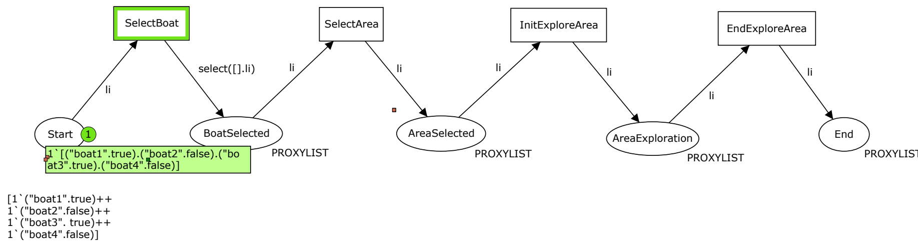

Fig. 1. Petri Net for the Explore Area Behavior. Transitions are represented as boxes and places as circles. Tokens represent a boat and whether such boat has been selected.

change the behavior of the exploration phase once such phase has started (i.e., when the tokens enter the AreaExploration place). In particular if one of the boat can not complete such phase the plan would not terminate correctly and the only choice that the operator has is to abort the plan.

The work we will present in this poster addresses this issue by providing a more refined mechanism to interact with the system. In particular, we define an interrupt model for the plans that allows the human operator to change the execution of a plan with the aim to recover from incorrect executions. In more detail, such mechanism provides the possibility to add two particular types of interrupt in a plan: a general one that involves all the boats and a *proxy* one that involves only a selected boat.

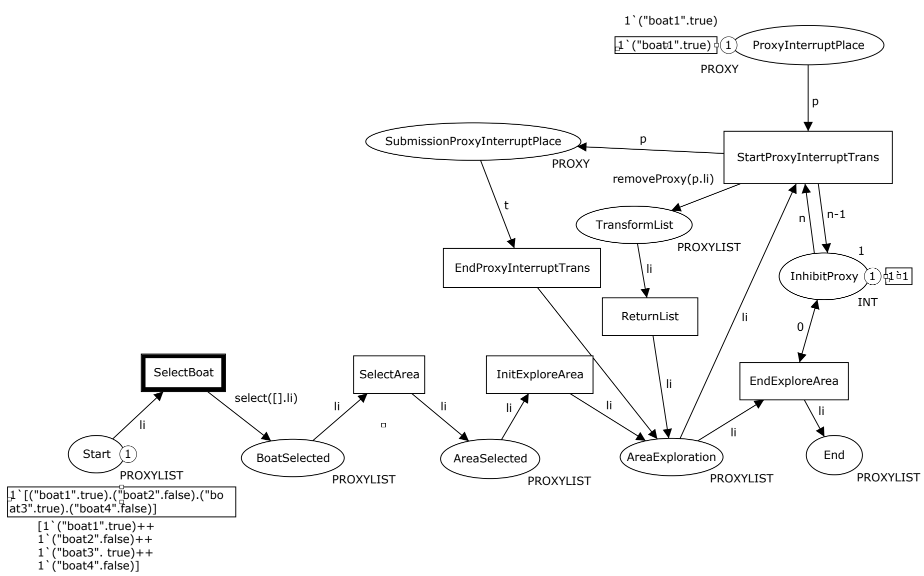

Fig. 2. Petri Net for the Explore Area Behavior with the interrupt mechanism.

Figure 2 provides a concrete example of a net with the addition of a *Proxy* interrupt that allows the human operator to intervene by executing a specific recovery procedure (represented by the SubmissionProxyInterruptPlace) in case the exploration can not be completed by one or more boats (i.e., the EndExploreArea transition does not fire). We used such mechanism on several plans for the CRW and executed such plans, showing that the interrupt mechanism allows the human operator to intervene while the plan is running allowing a sophisticate interaction with the system.

A second contribution of this work is to provide a mechanism to validate the plans by using a standard tool for Coloured Petri Net analysis, i.e. the CPNTools [4]. To do so, we correlate standard properties of Petri Net (such as dead markings, live markings and boundedness properties) to properties of the plan we are validating (i.e., whether a plan correctly reaches a final state). In particular, we focus on the analysis of the plan when specific actions may fail (e.g., the EndExploreArea) due to malfunctioning of the boats. As an example consider again the explore area plan in Figure 1. Our analysis builds the state space (i.e., all possible markings) of the net when no action fails. We query such state space looking for dead markings and we check whether all dead markings represent a final state for the net, if this is not the case the plan is not valid. We do the same analysis assuming one of the action may fail (to simulate this we inhibit the execution of such action with an inhibit state see Figure 2) and assuming that the interrupt will solve this problem (e.g., if a boat failed we remove the boat from the exploration mission). We then build the search space for this net and check whether this plan has dead marking that do not match with a final state. We performed such analysis on several plans for the CRW and we were able to show that in most cases the original plan would become invalid when an action fail and that the interrupt mechanism allows to correctly recover from such possible failures.

References

1. Delle Fave, F.M., Rogers, A., Jennings, N.R.: Argus: a coordination system to provide first responders with live aerial imagery of the scene of a disaster. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3, International Foundation for Autonomous Agents and Multiagent Systems (2012) 1467–1468
2. Scerri, P., Kannan, B., Velagapudi, P., Macarthur, K., Stone, P., Taylor, M., Dolan, J., Farinelli, A., Chapman, A., Dias, B., et al.: Flood disaster mitigation: A real-world challenge problem for multi-agent unmanned surface vehicles. In: Advanced Agent Technology. Springer (2012) 252–269
3. Ziparo, V., Iocchi, L., Lima, P., Nardi, D., Palamara, P.: Petri net plans. Autonomous Agents and Multi-Agent Systems **23**(3) (2011) 344–383
4. Ratzer, A.V., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K.: Cpn tools for editing, simulating, and analysing coloured petri nets. In: Applications and Theory of Petri Nets 2003. Springer (2003) 450–462