

RADAR - A Proactive Decision Support System for Human-in-the-Loop Planning

Sailik Sengupta and Tathagata Chakraborti and Sarath Sreedharan and Subbarao Kambhampati
{sailiks, tchakra2, ssreedh3, rao}@asu.edu
Arizona State University

Abstract

Proactive Decision Support (PDS) aims at improving the decision-making experience of *human decision makers* by enhancing the quality of the decisions and the ease of making them. Specifically, we propose a system RADAR that assists expert humans in the loop who now share a detailed, if not complete, model of the domain but may still be unable to compute plans due to cognitive overload. Our system supports naturalistic decision making by providing valuable suggestions and insights during the planning process rather than enforcing plans and executing actions by itself. We show, with the help of a firefighting domain, how RADAR helps the commander in validating, fixing and obtaining alternative plans. Furthermore, whenever the commander finds a suggested plan to be inexplicable, the system provides explanations, to increase the reliability of the automation system. The commanders reaction to explanations provides a learning opportunity for RADAR to update its model and account for these going forward. In this demo, we motivate the use of automated planning technologies in proactive decision support for expert humans, showcasing an end to end system that will prove to be a valuable test bed for upcoming research.

In complex decision-making environments, such as responding to crisis scenarios, high-level planning of effective responses is a key challenge (Laskey, Marques, and da Costa 2016). Although automated planning can play a key role in such situations, a set of challenges needs to be addressed for providing effective decision support.

Complete domain models of complex environments is difficult to obtain. Hence, plans generated by such systems cannot be relied upon completely. With RADAR, we support naturalistic decision making and provide explanations for suggesting a particular plan. We also account for the human's reactions to given explanations, using it as an opportunity to update our knowledge about the domain. Unlike "mixed-initiative planning", RADAR's goal is not to facilitate the humans to enter the land of the planner and help it, but rather enable the planner to do this for the human in the loop.

RADAR We provide details about the RADAR user-interface in Figure 1. In this section, we talk about the use cases it can support highlighting the planning technologies it uses to do so. We will use a fire-fighting scenario to illustrate the functionalities of our system. The domain model used by the system is represented in PDDL and is

assumed to be very close, if not identical, to that of the expert's model, which is also known. The scenario involves the local fire-fighting chief (the expert in the loop), who along with the local police, medical and transport authorities, is trying to build a plan in response to a fire using RADAR. A video demonstrating the system is available at <https://goo.gl/YunA21> and the domain and problem files can be found at <https://goo.gl/htrmLQ>.

Goal Selection and Focus. Once the fire chief selects a goal, RADAR generates the problem file and displays the landmarks (Hoffmann, Porteous, and Sebastia 2004) to help the user be on track to achieve the goal.

Plan Validation. RADAR does plan validation (Fox, Howey, and Long 2005) of a partial plan made by the fire chief and shows reasons as to why it is invalid.

Plan Correction and Plan Suggestions. When the commander has an invalid partial plan, he may ask RADAR to repair it. In addition, he may specify a partial plan and ask RADAR to suggest a complete plan respecting his partial plan. To do these, we leverage the compilation `pr2plan` from (Ramírez and Geffner 2010).

Explanation Generation. When the fire chief feels that a suggested plan is inexplicable, we can provide explanations based on domain model differences (Chakraborti et al. 2017). We also use this technology to solve the problem of finding a state from which the goal of a planning problem is reachable given an (initial) state from which it isn't.

Model Reconciliation. Based on whether the commander accepts the explanations given or not, we use this as a learning opportunity to update RADAR's model.

References

- Chakraborti, T.; Sreedharan, S.; Kambhampati, S.; and Zhang, Y. 2017. Explanation generation as model reconciliation in multi-model planning. Technical report.
- Fox, M.; Howey, R.; and Long, D. 2005. Validating plans in the context of processes and exogenous events. In *AAAI*, 1151–1156.
- Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *J. Artif. Int. Res.* 22(1):215–278.
- Laskey, K. B.; Marques, H. C.; and da Costa, P. C. G. 2016. *High-Level Fusion for Crisis Response Planning*. Cham: Springer International Publishing, 257–285.
- Ramírez, M., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *AAAI*.

Planning Panel

- Actions can be added, deleted and arranged in order.
- A partially built plan can be validated.
- Erroneous plans can be fixed.
- Plans can be suggested.
- RADAR can be asked for explanation of suggested plan.
- Changes made to the current plan can be un-done.

Goal selection panel

Once a goal is selected, the problem is created and predicate landmarks are shown.

Plan in Progress

(ADDRESS_MEDIA_TRANSPORTCHIEF) Add

Validate Fix Suggest Undo

Alternative Plan Explain this plan?

1 (CONTACT_MEDIA_POLICECHIEF)

(ALERT_FIRECHIEF_ADMINFIRE)

(DEPLOY_BIG_ENGINES_FIRECHIEF_ADMINFIRE_BYENG)

(EXTINGUISH_BIG_FIRE_FIRECHIEF_BYENG)

(ALERT_MEDICHIEF_ADMINFIRE)

(ALERT_FIRECHIEF_PHXFIRE)

(DEPLOY_HELICOPTERS_FIRECHIEF_PHXFIRE_BYENG)

(DEPLOY_BULLDOZERS_FIRECHIEF_ADMINFIRE_BYENG)

(ALERT_MEDICHIEF_ADMINFIRE)

(DEPLOY_RESCUERS_FIRECHIEF_ADMINFIRE_BYENG)

Goals

2

Select a goal:

Extinguish Big Fire At Byeng

Update Goal

addressed_media() extinguished_fire_byeng() needed_address_media() deployed_rescuers_byeng()
 has_rescuers_number_adminfire() alerted_adminfire() deployed_helicopters_byeng() alerted_phxfire()
 has_helicopters_number_phxfire() has_big_engines_number_adminfire() deployed_big_engines_byeng() fire_at_byeng()
 deployed_bulldozers_byeng() media_contacted_policechief()

3

View larger map

Phoenix Tempe Mesa Apache Junction

South Mountain Park AHWATUKEE FOOTHILLS

Fire Station's Resources

Fire Station	Small Engine	Big Engine	Ladders	Bulldozers	Helicopters	Rescuers
adminfire	✓	✓	✓	✓	✗	✓
mesafire	✗	✗	✓	✗	✗	✓
phxfire	✗	✗	✓	✓	✓	✓
scottsfire	✓	✗	✓	✓	✗	✓

4

Hospital's Resources

Hospital Ambulance Availability

joseph	✓
lukes	✗

Police Stations's Resources

Police Station	Police Cars	Policemen
apachestation	✗	✓
courtstation	✓	✗
substation	✓	✓

Map Panel

Shows location of the resources on the map and highlights the affected city.

Resource Panel

- Shows availability of resources at different locations.
- Clicking the cross and ticks updates resource status.
- Yellow lines indicate resource locations relevant to the current plan.

Figure 1: RADAR interface showing decision support for the human commander making plans in response to a fire.