RADAR-X: An Interactive Interface Pairing Contrastive Explanations with Revised Plan Suggestions

Valmeekam Karthik, Sarath Sreedharan, Sailik Sengupta, Subbarao Kambhampati
CIDSE, Arizona State University, Tempe, AZ 85281 USA
{kvalmeek, ssreedh3, sailiks, rao}@asu.edu

Abstract

Empowering decision support systems with automated planning has received significant recognition in the planning community. The central idea for such systems is to augment the capabilities of the human-in-the-loop with automated planning techniques and provide timely support to enhance the decision-making experience. In addition to this, an effective decision support system must be able to provide intuitive explanations based on specific queries on proposed decisions to its end users. This makes decision-support systems an ideal test-bed to study the effectiveness of various XAIP techniques being developed in the community. To this end, we present our decision support system `RADAR-X` that extends `RADAR` (Sengupta et al. 2017) by allowing the user to participate in an interactive explanatory dialogue with the system. Specifically, we allow the user to ask for contrastive explanations, wherein the user can try to understand why a specific plan was chosen over an alternative (referred to as the foil). Furthermore, we use the foil raised as evidence for unspecified user preferences and use it to further refine plan suggestions.

Introduction

Proactive decision support systems are a case of human-in-the-loop planning (Kambhampati and Talamadupula 2015) where the human is responsible for making the decisions and is supported by an automated planning system in complex decision-making scenarios. Such systems have been shown to aid the user in making faster and better decisions (Grover et al. 2019). Given that the human (whom we assume to be an expert) is responsible for the final plan in this mixed-initiative setting, a key aspect required for the success of this synergy is to support the user's requirement for explanations, especially when the suggestions made by the system are not acceptable to the user. Previously, works on decision support systems such as `RADAR` (Sengupta et al. 2017) have considered technologies in Explainable AI Planning (XAIP) (Chakraborti et al. 2017; Sreedharan, Kambhampati, and others 2018) to facilitate such interaction but the participation of the user in the explanatory dialogue was limited. `RADAR` provided minimally complete model reconcilation explanations (presented in (Chakraborti et al. 2017)) as and when required but the explanations were not based on specific user queries. This might lead to the system providing more verbose explanations which can be hard for the decision maker to understand. To avoid such situations, the system should let the decision-maker drive the explanatory dialogue and provide explanations that cater to the specified query. In this regard, we propose `RADAR-X`, an extension of the `RADAR` system (Sengupta et al. 2017), that supports interactive contrastive explanations (Miller 2018) and uses it as the main vehicle for the interaction between the system and the user. We enable users to specify alternatives (referred to as foils) to a plan suggested by the system and ask for explanations that cater to the specified foil. Moreover, we look at the foils as a specification of the user's latent preferences and use that interpretation to come up with refined plan suggestions. We have also made some interface improvements that aid in smoothening the interaction process.

To generate explanations based on user-specified foils, we adapt the model reconcilation framework presented in (Chakraborti et al. 2017). In particular, when the user provides a partial-plan as a foil (representative of a set of expected plans) that is invalid in the system's true model, the system can generate an explanation that refutes the foil. A partial-plan can informally be seen as a set of actions with ordering constraints that represent a set of potential solutions to a planning problem (Kambhampati, Knoblock, and Yang 1995). Along with providing foil-based explanations, inspired from ideas in iterative planning (Smith 2012), we consider the case of proactive preference elicitation where plan suggestions are refined based on the specified foil, which are an indication of the user's latent preferences. We look at three different interaction strategies through which plan suggestions are refined– (1) we develop a novel encoding to generate the nearest plan to the specified foil which implies using the largest possible part of the foil, (2) we employ a search through the space of subsets of the foil to find sets containing conflicting actions and present these sets for the user to resolve, thereby eliciting their preference to generate a plan suggestion and (3) we look to present all the maximal plausible subsets of the foil to the user as options to choose from and use that selection to provide a plan that the user prefers.

In this paper, we start by giving a background of the explanations paradigm that we will leverage. Then, we provide an overview of the implemented interface and detailed illustration of the domain. With the help of use-cases, we showcase the two technical problems that can be addressed by our system. Finally, we discuss the design principles laid out in the Human-Computer Interaction (HCI) literature that our system follows.

Background

In this section, we provide a quick overview of topics in automated planning, necessary to understand the proposed techniques.

A **Classical Planning Problem** can be described as a tuple $\mathcal{M} = \langle \mathcal{D}, I, G \rangle$, consisting of a domain $\mathcal{D} = \langle F, A \rangle$ where F is a finite set of fluent symbols that define a state $s \subseteq F$ and A corresponds to a finite set of actions and $I, G (\subseteq F)$ represent the initial and goal states. An action $a \in A$ is associated with a cost c_a, a set of preconditions $pre(a) \subseteq F$ and a set of effects $eff(a) \subseteq F$. These effects can be further separated into a set of add effects $eff^+(a)$ and a set of delete effects $eff^-(a)$. The action $a \in A$ can be represented as a tuple $\langle c_a, pre(a), eff^+(a), eff^-(a) \rangle$ and can only be executed in a state s if $s \models pre(a)$ i.e., $\delta_{\mathcal{M}}(s, a) \models s \cup eff^+(a) \setminus eff^-(a)$ *if* $s \models pre(a)$; *else*, $\delta_{\mathcal{M}}(s, a) \models \perp$ where $\delta_{\mathcal{M}}(.)$ is the transition function. The solution to such a problem is a plan π defined as a sequence of actions $\langle a_1, a_2, ..., a_n \rangle$ such that $\delta_{\mathcal{M}}(I, \pi) \models G$ and $\delta_{\mathcal{M}}$ here is a cumulative transition function given by $\delta_{\mathcal{M}}(s, \langle a_1, a_2, ..., a_n \rangle) = \delta_{\mathcal{M}}(\delta_{\mathcal{M}}(s, a_1), \langle a_2, ..., a_n \rangle)$. A sequence of actions that has an unmet precondition and thus, cannot achieve the goal has cost ∞. On the other hand, the cost of a plan π is the sum of the costs of all the actions present in the plan and is given by $C(\pi, \mathcal{M}) = \sum_{a \in \pi} c_a$. The cost of the optimal plan π^* is denoted as $C^*_{\mathcal{M}}$ where $\pi^* = \arg\min_\pi \{C(\pi, \mathcal{M}) \forall\, \pi \text{ where } \delta_{\mathcal{M}}(I, \pi) \models G\}$.

In classical planning, the human is considered to have the same planning model and reasoning capabilities as the planner but often, the human's understanding may significantly differ from that of the planner. The lack of a common ground is more likely to exist in the context of naturalistic decision making scenarios characterized by (1) cognitive overload and (2) lack of situational awareness. Thus, we can view this as a **Multi-Model Planning (MMP)** scenario where $\mathcal{M}^R = \langle \mathcal{D}^R, I^R, G^R \rangle$ is the planner's model of the planning problem and $\mathcal{M}^H = \langle \mathcal{D}^H, I^H, G^H \rangle$ is the human's understanding of the same. Note that the best possible plan π generated by the system, evaluated in terms of a different model that is in the human's mind, may cease to be optimal in such settings, i.e $C(\pi, \mathcal{M}^R) = C^*_{\mathcal{M}^R}$, but $C(\pi, \mathcal{M}^H) > C^*_{\mathcal{M}^H}$. The difference between these two models, thus, becomes a key factor in the explanation setting. The system tries to achieve common ground with the human by bringing the human's model closer to its own model through explanations in the form of model updates. This is formalized as a Model Reconcilation Problem in (Chakraborti et al. 2017). Some works consider a different viewpoint of the planner, where the planner accounts for human intents and changes its behavior to be explicable (Zhang et al. 2017; Kulkarni et al. 2016) or legible (Dragan, Lee, and Srinivasa 2013) to the human. But applying these works directly in the context of decision support might not enable a user to interact and refine their preferences which is considered as a key aspect in decision support systems.

A Model Reconciliation Problem (MRP), as defined in (Chakraborti et al. 2017), can be represented using the tuple $\langle \pi^*, \langle \mathcal{M}^R, \mathcal{M}^H \rangle \rangle$ where π^* is the optimal plan in $\mathcal{M}^R$ $(C(\pi^*, \mathcal{M}^R) = C^*_{\mathcal{M}^R})$. MRP is constructed as a model-space search whose solution is considered as an explanation consisting of a set of model updates. Of the four types of explanations defined in (Chakraborti et al. 2017), `RADAR` considers the minimally complete explanations (Sengupta et al. 2017), which we now define.

A **Minimally Complete Explanation (MCE)** is the minimal set of relevant information that is provided to the human to explain the optimality of the plan. The objective here is to find the minimum number of differences between the human's model ($\mathcal{M}^H$) and the planner's model ($\mathcal{M}^R$) such that the plan in the planner's model is optimal in the updated human's model. It is given by $\mathcal{E}^{MCE} = \arg\min_{\mathcal{E}} |\Gamma(\widehat{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^H)|$ with $C(\pi^*, \widehat{\mathcal{M}}) = C^*_{\widehat{\mathcal{M}}}$ where $\Gamma(\mathcal{M})$ denotes a mapping function that represents a planning problem $\mathcal{M} = \langle \langle F, A \rangle, \mathcal{I}, \mathcal{G} \rangle$ as a state in the space of models defined by model parameters $\mathcal{F}$. Formally,

$$\Gamma(\mathcal{M}) = \{\tau(f) | \forall f \in I \cup G$$
$$\bigcup_{a \in A} \{f' | \forall f' \in \{c_a\} \cup \{pre(a)\} \cup \{eff^+(a)\} \cup \{eff^-(a)\}\}\}$$

where $\tau(f)$ is as follows:

$$\tau(f) = \left\{ \begin{array}{ll} \textit{init-has-f} & \textit{if } f \in I \\ \textit{goal-has-f} & \textit{if } f \in G \\ \textit{a-has-precondition-f} & \textit{if } f \in pre(a), a \in A \\ \textit{a-has-add-effect-f} & \textit{if } f \in eff^+(a), a \in A \\ \textit{a-has-del-effect-f} & \textit{if } f \in eff^+(a), a \in A \\ \textit{a-has-cost-f} & \textit{if } f = c_a, a \in A \end{array} \right\}$$

and the set of model parameters is given by:

$$\mathcal{F} = \{\textit{init-has-f} | \forall f \in F^H \cup F^R\} \cup \{\textit{goal-has-f} | \forall f \in F^H \cup F^R\}$$
$$\bigcup_{a \in A^H \cup A^R} \begin{array}{l} \{\textit{a-has-precondition-f}, \textit{a-has-add-effect-f}, \\ \textit{a-has-del-effect-f} | \forall f \in F^H \cup F^R\} \end{array}$$
$$\cup \{\textit{a-has-cost-}c_a | a \in A^H\} \cup \{\textit{a-has-cost-}c_a | a \in A^R\}.$$

MCE, although time-consuming to compute, tries to decrease the human's cognitive load by reducing the amount of irrelevant information provided as part of an explanation. Thus, in this work, we try to adapt the objective of MCE for generating contrastive explanations. Before we elaborate on that, we first delve into the interface of `RADAR-X` and describe the use-case that we support.

Figure 1: 'Add foil' page where users can add actions to their foil and ask for explanations or refined plan suggestions based on the specified foil.

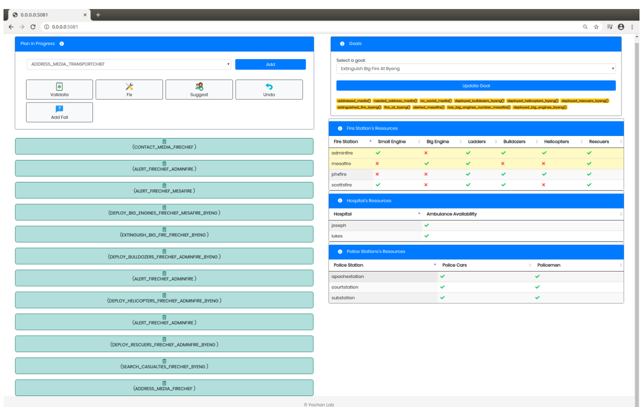

Figure 2: The interface of `RADAR-X` with various decision supporting functionalities for a human commander making plans in response to a fire.

`RADAR-X`

In (Sengupta et al. 2017), the authors consider a fire fighting scenario where `RADAR` is helping a fire-fighting chief, along with several other authorities, build a plan to extinguish a fire at a particular city in Arizona, USA. This scenario was represented as a classical planning problem in the PDDL (McDermott et al. 1998). We use the same domain to illustrate the capabilities of `RADAR-X`. We assume that the system has a model of the task ($\mathcal{M}^R = \langle \mathcal{D}^R, I^R, G^R \rangle$) that may be different from the human's model ($\mathcal{M}^H = \langle \mathcal{D}^H, I^H, G^H \rangle$) but $\mathcal{M}^H$ is known to the system R beforehand. Also, we assume $I^H = I^R$ and $G^H = G^R$.

Overview of the Interface

`RADAR-X` seeks to extend the capabilities of `RADAR` by allowing the user to engage in an explanatory dialogue with the system. Once a plan is suggested by the system, users can utilize the `Add Foil` button present in the `Plan in progress` panel shown in Figure 2 and are redirected to a new page shown in Figure 1. On this page, users can provide their foil that is a set of specific actions and ordering constraints over these actions; it may be a partial plan. Let us look at the three panels presented on this new page (as shown in Figure 1).

1. **Foil Panel**: This is the pivotal panel of `RADAR-X`. This panel allows users to specify foils; it provides the user with the ability to (1) add/delete actions from the foil, (2) change the ordering of the actions present in the foil and (3) choose from options when the foil cannot result in a feasible plan (such as asking for explanations or refined plan suggestions). These options in the panel will be our primary focus in the upcoming sections.
2. **Foil through Speech**:
   - **Speech Panel:** Instead of choosing actions from a drop-down list in the Foil panel, this panel facilitates users to specify foils using natural language. For example, a user can say, "Why contact media firechief and why not contact media transportchief?" to replace the CONTACT_MEDIA_FIRECHIEF action in the suggested plan with the expected action of CONTACT_MEDIA_TRANSPORTCHIEF.
   - **Action Transcript Panel:** Once the user specifies the foil using natural language, this panel displays the transcript of their speech; it seeks confirmation before replacing the present plan with the specified foil. This is

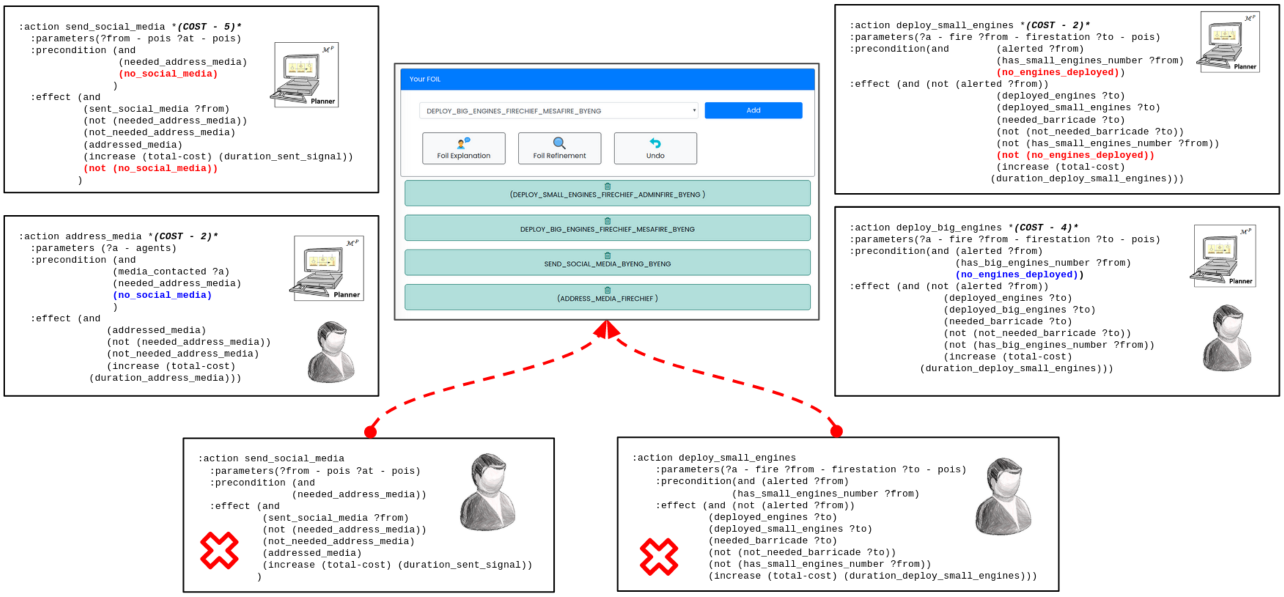

Figure 3: The given foil is raised due to incorrectness in the human's model.

mostly done to ensure that the inaccuracies in speech recognition technology do not result in the addition of unidentifiable actions. For now, we assume that complete action names are specified when this modality is used and believe that accurate parsing of a user's unrestricted language to foils will be an interesting and helpful future work.

3. **Current Plan Panel**: This panel acts as a reference point for the user. While the user works on the specified foils, the panel lists the original plan suggested by the system.

Use-Case Description

As mentioned earlier, the planning problem considered requires a fire-chief, with a decision support system by its side, to come up with a plan for extinguishing a fire in the city of Tempe. In this domain, there are two possible goals– extinguishing a big fire *vs.* extinguishing a small fire. Let us assume that the goal of the user here is to extinguish a small fire. When RADAR−X suggests a plan (as shown in the *Current Plan* panel in Figure 1) that does not meet the expectations of the user (due to differences between $\mathcal{M}^R$ and $\mathcal{M}^H$), the user can specify a foil (that may be a partial plan) and ask for explanations. In our example, shown in Figure 3, there are two specific parts in the suggested plan that are incongruous with the user's expectation. First, the user wants to send out information via a social media post (using SEND_SOCIAL_MEDIA) alongside addressing the media themselves (using ADDRESS_MEDIA_FIRECHIEF), but they are not aware that once posted on social media, the media will pick up the news from these forums, thus not requiring the fire-chief to separately address them. Technically, this is expressed as a delete effect of SEND_SOCIAL_MEDIA called NO_SOCIAL_MEDIA that is also a precondition for ADDRESS_MEDIA_FIRECHIEF in $\mathcal{M}^R$. This delete effect is missing in $\mathcal{M}^H$, leading them to believe that both are possible together. Second, the user expects that big engines should be deployed along with small engines, i.e. the actions DEPLOY_SMALL_ENGINES_FIRECHIEF_ADMINFIRE_BYENG and DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG should both be present in the plan. Here, the user is unaware that both types of engines cannot be deployed together and there is a delete effect of DEPLOY_SMALL_ENGINES_FIRECHIEF_ADMINFIRE_BYENG called NO_ENGINES_DEPLOYED that is a precondition for DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG. Therefore, when RADAR−X presents the plan suggestion, the user does not expect that only one of each of the expected action pairs, i.e. ADDRESS_MEDIA_FIRECHIEF and DEPLOY_SMALL_ENGINES_FIRECHIEF_ADMINFIRE_BYENG, appears in the suggested plan. The user thus raises a foil π' containing the four actions in the given order.

$\pi' =$

– DEPLOY_SMALL_ENGINES_FIRECHIEF_ADMINFIRE _BYENG

– DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG

– SEND_SOCIAL_MEDIA_BYENG_BYENG

– ADDRESS_MEDIA_FIRECHIEF

In this case, the actions SEND_SOCIAL_MEDIA and DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG can be viewed as the latent preferences of the user which are exposed when the user raises the foil to ask for explanations. We will now look at how RADAR−X generates (1) explanations to refute the given foil and (2) elicit the preferences of the user through three different interaction strategies.

Supporting Contrastive Explanations

A **Contrastive Explanation** answers the questions "Why P and not Q?" where P is the fact (the suggested plan) being explained and Q is the foil (the alternative proposed by the explainee) (Miller 2019). In (Chakraborti et al. 2017), the explanation techniques used in RADAR (Sengupta et al. 2017), the explanation answers the question "Why π?" where π is the suggested plan. Although this can be viewed as an implicit contrastive query, as in "Why π as opposed to any other plan $\pi'(\neq \pi)$?", it might make the explanation unnecessarily verbose when the user's expected set of plans is much smaller than the set of all optimal plans. Therefore, in RADAR-X, we let the user specify their expected set of plans as partial foils and empower the system to generate focused explanations that establish how the current plan compares against their specified set of plans. RADAR-X's contrastive explanation allows the system to reduce the amount of information provided and, in turn, reduce the human-in-the-loop's cognitive load.

In this setting, we will focus on scenarios where the mismatch between the suggested plan and the foil, which is presented as a partial plan, arise due to model mismatch. As mentioned earlier, the user's foil represents a set of actions and ordering constraints over these actions; it can be thought of as specifying a set of alternate plans, where every plan in the set includes the specified actions and meets the ordering constraints. A case for explanation arises when the specified foil (1) cannot be part of a (valid) plan in $\mathcal{M}^R$ or (2) is sub-optimal or costlier than the optimal plan suggested by the system. We consider the former case and, further, allow users to raise additional foils after each explanation thereby making the explanation a multi-step procedure. If $\hat{\Pi}$ is the set of alternate foils raised by the human, the objective of minimally complete explanation, in our setting, is to start from $\mathcal{M}^H$ and reach a goal state $\widehat{\mathcal{M}}$ where no foil in $\hat{\Pi}$ can result in a (valid) plan in $\widehat{\mathcal{M}}$ (using a subset of the model differences, which we denote by $\mathcal{E}$, between $\mathcal{M}^H$ and $\mathcal{M}^R$). Formally, a minimally contrastive explanation can be expressed as follows.

$$\mathcal{E}^{con} = \arg\min_{\mathcal{E}} |\Gamma(\widehat{\mathcal{M}})\Delta\Gamma(\mathcal{M}^H)|$$
$$s.t. \qquad C(\pi', \widehat{\mathcal{M}}) = \infty \qquad\qquad \forall \pi' \in \hat{\Pi}$$

RADAR-X adapts the MCE search in (Chakraborti et al. 2017) to satisfy this objective and come up with the required explanation.[1] The explanation $\mathcal{E}$ can be viewed as the correction that needs to be made in the human model for refuting the suggested foil.

For the use-case described above, RADAR-X, given the foil, generates an explanation where the action SEND_SOCIAL_MEDIA_BYENG_BYENG deletes an effect NO_SOCIAL_MEDIA which is a precondition of ADDRESS_MEDIA_FIRECHIEF (see Figure 4). Hence, the given foil is invalid in the planner's model. This explanation is the minimal explanation required to refute the foil. Once the explanation is presented, the human's model can then be corrected by adding the delete effect into the model.

[1]Note that the goal check condition requires us to ensure that, given the foil, no valid plan can be realized in $\widehat{\mathcal{M}}$. We can rely on faster unsolvability tests that are sound but not complete. For example, in this work, we used h^m based tests in the initial states. Additionally, to ensure completeness, we iteratively increased the value of m after each run until we found a solution. Also, our initial value of m was 1.

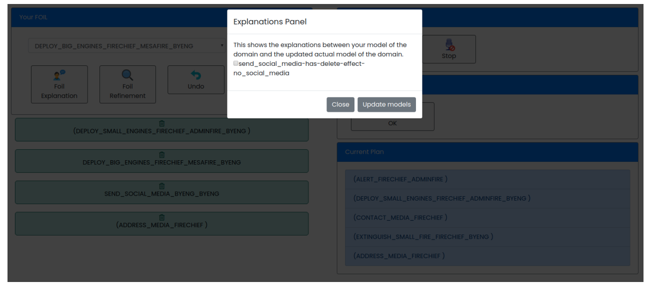

Figure 4: The minimally complete explanation generated states that the action SEND_SOCIAL_MEDIA has a delete effect NO_SOCIAL_MEDIA.

Proactive Preference Elicitation– Suggesting Plans

Even though the human's model is updated and the human understands that the given foil is invalid, the foil serves a second purpose– it is indicative of some latent preferences of the user. We hypothesize that asking for contrastive explanations exposes some of the preferences that the user does not specify explicitly. Thus, one can use foils to identify plans that are closer to the human's expectations. This can be seen as a problem of constrained plan revision (Talamadupula, Smith, and Kambhampati 2014). We use three different approaches in RADAR-X to identify such plans.

The Closest Plan approach In this approach, we look at generating the *closest plan* to the specified foil which implies using the largest part of the foil in the revised plan. For this, we revisit the plan-recognition-as-planning methodology presented in (Ramírez and Geffner 2009) and construct a simple yet effective compilation that encodes the partial foils as soft constraints and imposes penalties if any of them are violated when coming up with a plan. In terms of plan recognition, one can view this as the system trying to recognize plans that are in the user's mind by leveraging the observations (which is the foil) presented by the user. In this case, the original plan in the user's mind cannot be realized as it is based on an incorrect understanding of the task. Hence, the system tries to come as close as possible to the expected plan by inducing soft constraints into the original planning problem, transform it into a new problem and generate a plan based on this new problem. While our algorithm is similar in spirit to (Sohrabi, Riabov, and Udrea 2016), the objective there is to allow for unreliable (noisy or missing) observations through the relaxation of the plan recognition as planning formulation whereas in our case, we consider the observations specified by the user to be perfect and look to utilize the maximum number of observations to generate a plan closest to the foil. Moreover, they consider observations over fluents rather than actions.

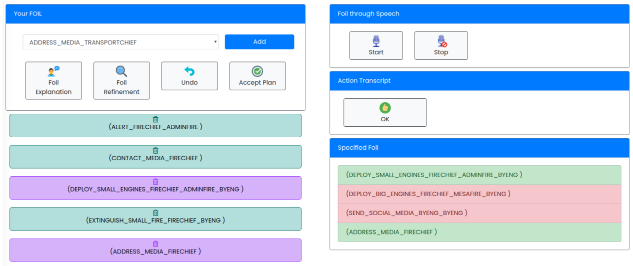

Figure 5: Generating the closest plan that uses the largest part of the foil.

A **Plan Recognition Problem** is represented by a tuple $R = \langle \mathcal{D}, I, O, \mathcal{G} \rangle$, where $\mathcal{D} = \langle F, A \rangle$ is the planning domain, $I \subseteq F$ is the initial state, $\mathcal{G}$ is the set of possible goals G, $G \subseteq F$, and $O = \langle o_1, o_2, ..., o_n \rangle$ is an observation sequence with each o_i being an action in A and $i \in [1, n]$. We compile the foil to a new planning problem by augmenting the existing actions in the original problem with a set of 'explain' and 'discard' actions for each observation o_i present in the observation sequence O. Here, the observations represent actions in the partial foil that the user specifies. An 'explain' action for an observation o_i is a replica of the observation with an additional effect (met_{o_i}) that indicates the observation is met (i.e. action o_i in the foil is used). On the other hand, a 'discard' action for an observation o_i is a dummy action which only has the effect (met_{o_i}) but has a cost that is significantly higher than the corresponding 'explain' action. This means discarding any observation should be costlier. For now, we look to preserve the ordering constraints of the observations by adding $met_{o_{i-1}}$ as a precondition for the observation o_i with $i \in [2, m]$.

A **Transformed Planning Problem** for a plan recognition problem $R = \langle \mathcal{D}, I, O, \mathcal{G} \rangle$ is $P' = \langle \mathcal{D}', I', G' \rangle$ where $\mathcal{D}' = \langle F', A' \rangle$ and:

- $F' = F \cup \{met_{o_i} | o_i \in O\}$,
- $I' = I$
- $G' = \{g | \forall g \in \mathcal{G}\} \cup \{met_{o_i} | o_i \in O\}$
- $A' = A \cup A_{explain} \cup A_{discard}$
  - $A_{explain} = \{e_{o_i} | o_i \in O,\ c_{e_{o_i}} = c_{o_i},\ pre(e_{o_i}) = \{pre(o_i) \cup met_{o_{i-1}}$ if $i > 1$; else, $pre(o_i)\},\ eff^+_{e_{o_i}} = \{eff^+_{o_i} \cup met_{o_i}\},\ eff^-_{e_{o_i}} = \{eff^-_{o_i}\}\}$
  - $A_{discard} = \{d_{o_i} | o_i \in O,\ c_{d_{o_i}} >> c_{e_{o_i}},\ pre(d_{o_i}) = \{\emptyset\},\ eff^+_{d_{o_i}} = \{met_{o_i}\},\ eff^-_{e_{o_i}} = \{\emptyset\}\}$

Using this transformed planning problem, the planner generates a plan which uses the largest possible part of the foil. In the previously mentioned example, the generated plan contains two of the actions presented in the partial foil π'. The used actions ADDRESS_MEDIA_FIRECHIEF and DEPLOY_SMALL_ENGINES_FIRECHIEF_ADMINFIRE_BYENG are encoded in a different color to help the user to easily identify parts of the foil used (see Figure 5). Additionally, the interface replaces *'Current Plan'* panel with the *'Specified Foil'* Panel (at the bottom right of Figure 5) where the actions used (observations met) in the generated plan are encoded in green color and actions discarded (unmet observations) are encoded in red. These actions can be added or deleted by clicking on the action in the panel. Further, we allow the users to directly add preferred actions that are not present in the generated plan, specify another foil, and engage in a longitudinal interaction.

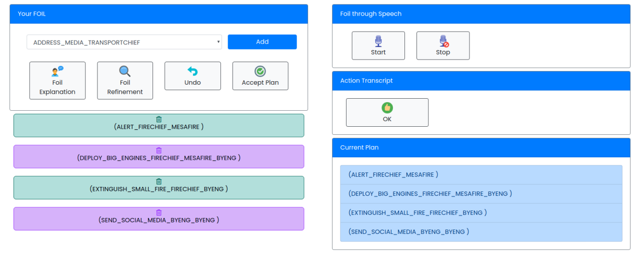

Figure 6: Conflict-set resolution to generate preferred plans.

The Conflict Sets approach Even though the plan generated using the above compilation utilizes the largest part of the foil, the actions may have different importance to the user; hence, a planner may choose to use parts of the foil that are less important to the user. For example, the two actions from the specified foil (ADDRESS_MEDIA_FIRECHIEF and DEPLOY_SMALL_ENGINES_FIRECHIEF_ADMINFIRE_BYENG) that were present in the generated plan may be less preferred by the user than the actions that were discarded (SEND_SOCIAL_MEDIA_BYENG_BYENG and DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG). Thus, to reach the final plan that the user prefers, they might have to engage in recurring interactions. This increases the amount of effort the user has to put in to make sure that the planner generates a plan of his/her liking. A simple attempt to reduce the cognitive load on the user would be to provide all possible sets of conflicting actions in the specified foil and ask the user to resolve them. To find such conflict sets, we employ a systematic breadth-first search in the space of subsets of the foil. The idea here is similar to that of the Systematic Strengthening (SysS) approach in (Eifler et al. 2020). Starting from the empty set, for each subset of the specified foil, we use the compilation specified in (Ramírez and Geffner 2009) to compile it into a planning problem and check whether the compiled problem is unsolvable.[2] If the planning problem is unsolvable, the corresponding subset would be deemed as a conflict set and the user would have to resolve by selecting one action to remove from the subset. In the case of singleton subsets, we provide the user with a message that including the action cannot result in a valid completion. Note that if the empty set is deemed as a conflict set, it implies that the original planning problem itself was unsolvable. Such a case cannot arise in RADAR-X because the start of the interaction (which is

[2] If we forget the need to get optimal conflict sets, we can rely on faster unsolvability tests that are sound but not complete; similar to the one mentioned previously.

based on a proposed plan) ensures that the planning problem is solvable. Also, as mentioned earlier, we look to preserve the ordering constraints for now, hence, we do not consider the permutations of a subset; but this can be relaxed as well. Once the conflict sets have been resolved by the user, the system can generate a plan that contains the preferred actions specified in the foil (that are also non-conflicting) along with the actions that are not part of any conflict set. While this method helps to elicit the user's preferences, the space of the subsets of the foil, even barring permutations, explodes combinatorially; thus, calculating the conflict sets can be expensive.

In our use-case, `RADAR-X` searches all the subsets of the partial foil, starting from the empty set. (DEPLOY_SMALL_ENGINES_FIRECHIEF_ADMINFIRE _BYENG, DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG) is a conflict set as the predicate NO_ENGINES_DEPLOYED gets deleted by the former action and is required as a precondition for the latter one. This is presented to the user to elicit the user's preferred action. Then, the next conflict set presented to the user is (SEND_SOCIAL_MEDIA_BYENG_BYENG, ADDRESS_MEDIA_FIRECHIEF). Using both the preferences the final plan is generated and the preferred actions are encoded in a different color for the user to identify.[3] This is shown in Figure 6. Note that some of the conflict sets presented to the user may not be resolvable in the human's model without additional explanations. As a next step, we look to allow for providing such explanations.

The Plausible Sets approach Instead of making the user resolve conflict sets, the system can also present the plausible subsets of the foil as options for the user to choose from. In this approach, we aim to present all the maximal valid subsets of the foil to the user. Maximal valid subsets can be considered as subsets which contain the maximum number of actions from the foil and a plan can be generated using *all* of the actions present in the subset. To find such sets, we use an idea similar to that of Systematic Weakening (SysW) mentioned in (Eifler et al. 2020). Starting from the entire foil as a set, for each subset, we compile it into a planning problem (using the compilation specified in (Ramírez and Geffner 2009)) and try to generate a plan. An unsolvability test (similar to that of the previous approach) is done before generating the plan to discard subsets that are found to be unsolvable. In the case of successful plan generation, the corresponding subset is deemed to be valid. Note that subsets that are already part of a valid subset are not checked as we aim to present the maximal plausible sets. Once all the valid subsets are found, they are presented to elicit the preference of the user and then, based on the preference, a plan is suggested. Similar to the conflict sets approach, this would also be an expensive computation as the search is in the space of subsets.

`RADAR-X` presents all the maximal plausible subsets for the previously illustrated example (as shown in Figure 7) from which the user can choose the preferred

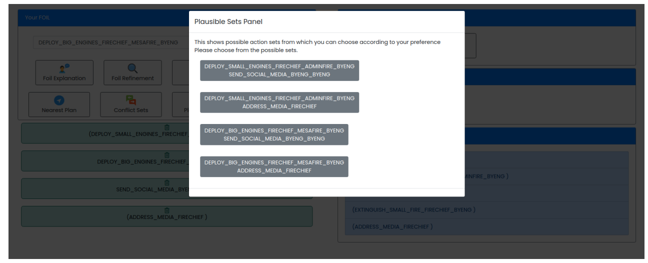

Figure 7: Presenting maximal plausible sets to generate preferred plans.

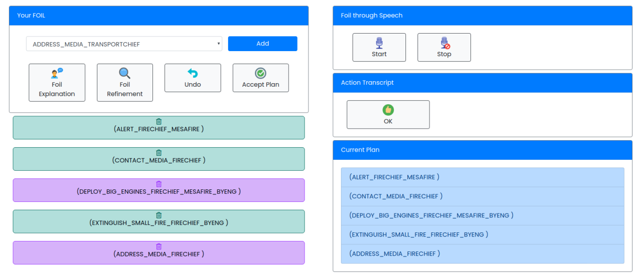

Figure 8: The generated plan if the user chooses the subset (DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG and ADDRESS_MEDIA_FIRECHIEF).

one. For instance, if the user chooses the subset (DEPLOY_BIG_ENGINES_FIRECHIEF_MESAFIRE_BYENG and ADDRESS_MEDIA_FIRECHIEF), Figure 8 shows the plan that is generated.

System Design

In `RADAR` (Sengupta et al. 2017), the authors propose a decision support system and carefully situate their systems' capabilities along two independent axes defined in Human-Computer Interaction (HCI). The first axis enumerates the ten levels of automation (Sheridan and Verplank 1978). As shown in Figure 9, each point along this axis represents the extent of automation a system supports– the lowest point indicate minimal support (where the human does most of the work) and the highest point indicates complete automation (where the system does the work and informs the human if it thinks they should be told). The second axis divides a system's capabilities into four stages - Information Acquisition, Information Analysis, Decision Selection, and Action Implementation - and define the levels of automation quasi-independently across these stages (Parasuraman 2000). In this work, we focus on two stages– Information Analysis (providing explanations) and Decision Selection (refining plan suggestions)– and situate the capabilities of `RADAR-X` on the automation ladder with respect to these stages in Figure 9.

1. **Contrastive Explanations**: After `RADAR-X` presents the explanations, if the user truly believes that the planner's model is the faulty one, an option to veto the model update is given to the user where they can choose to disap-

[3]For our example, in the unsolvability tests with the h^m preprocessor, m=1 found the solution.

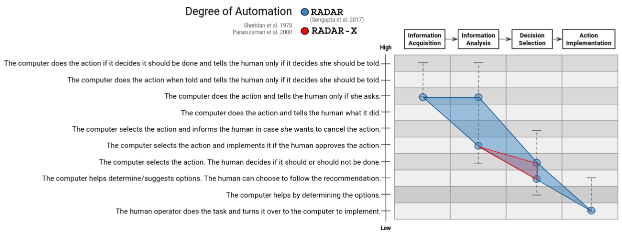

Figure 9: The automation ladder showing various stages of decision support and the role of `RADAR` and `RADAR-X` in it

prove the explanations provided by the system, in turn, asking the system to update $\mathcal{M}^R$. This sort of an interaction places the information analysis part of our system at degree five of automation.

2. **Proactive Preference Elicitation**: Providing plan suggestions to the user falls under the decision selection stage in the automation hierarchy. In providing the closest plan, `RADAR-X` essentially provides a plan that utilizes the maximum possible part of the specified foil and the user decides whether to accept it or further engage in recurring interactions. Hence, `RADAR-X` demonstrates degree four of automation here. Presenting the conflict sets or maximal plausible subsets narrows down the options for the user to select from, which places the system at degree three of automation.

We carefully position our system in the lower part of the automation ladder because ascending the automation ladder, which may improve the operational performance, can drastically decrease the response to failures or mistakes made by the system (Wickens et al. 2010). The latter becomes crucial in decision support scenarios as the human is held responsible for the final plan.

Related Work

The spectrum of work in human-in-the-loop planning (Kambhampati and Talamadupula 2015) ranges from the more traditional mixed-initiative settings (Ferguson et al. 1996; Ai-Chang et al. 2004; Kim, Banks, and Shah 2017) where the planners drove the interaction in these scenarios with the users 'advising' them, to works in decision support systems (Sengupta et al. 2017; Sengupta, Chakraborti, and Kambhampati 2018; Mishra et al. 2019) where the user is responsible for the plan while the system provides support. In (Sengupta et al. 2017), the authors propose a proactive aspect to decision support systems and design the system's capabilities based on principles in Human-Computer Interaction (HCI). In (Grover et al. 2019), the authors show that such systems can improve the efficiency of decision making, the quality of the decision made, and increase user-satisfaction. Our work builds upon existing work enabling the user to specify foils, refute them with explanation and engage in a discussion until consensus is reached.

In XAIP, several works have considered the problem of generating contrastive explanations (Chakraborti, Sreedharan, and Kambhampati 2020). While we assume the user, who is held responsible for the final plan, is an expert user, (Sreedharan, Srivastava, and Kambhampati 2018) relaxes this assumption and consider searching a model lattice of possible abstractions of the planner's model to generate contrastive explanations that are appropriate to the level of the user's understanding of the task. Further, many works view contrastive explanations as identifying exemplary plans that satisfy the constraints given by the user (Krarup et al. 2019; Cashmore et al. 2019). In these cases, the explanation is based on the differences between the computed plan and the exemplary plan. In (Eifler et al. 2020), the authors take a new approach at addressing contrastive questions in oversubscription planning; they utilize plan-property entailments where the explanation for not achieving a certain property is in terms of a set of plan properties that are entailed by the given property. They can also use these entailments to provide *global* explanations. The idea of generating conflict sets as a means of eliciting the user's preferences stemmed from (Junker 2001), where preferred conflicts and relaxations are computed in a constraint satisfaction setting to explain the failure of a constraint solver.

Conclusion and Future Work

In this article, we presented `RADAR-X`, a decision-support system that looks to establish an interactive explanatory dialogue with the user. We looked at the two major technical aspects of this new interface. One enabled the user to ask for contrastive explanations by specifying a foil; this helps the user understand why the plan suggested by the system was chosen over the alternative. `RADAR-X` also refines plan suggestions using the specified foil as a stand-in for the user's latent preferences. This is done by (1) providing a *closest plan* or (2) presenting *conflict sets* or (3) *maximal plausible sets* within the foil thereby eliciting the user's preferences. If all the actions specified in the foil are not equally important to the user, we proposed two approaches, one based on conflict-set selection to present all the conflicting subsets of the foil, and the other to present the maximal plausible subsets of the foil. Although search in the space of action subsets makes it computationally heavy, it provides an approach to elicit the user's latent preferences in decision support scenarios.

In conclusion, providing contrastive explanations and refining plan suggestions complement each other in this scenario as the actual preference (foil) of the user cannot be satisfied and an explanation would help the user to better understand the foil's invalidity and trust the system better (Mercado et al. 2016) while refining the suggestions can be seen as an attempt to satisfy the user to the maximum extent possible using the foil as an indication of the user's latent preferences. It would be interesting to test the efficacy of the explanations generated and understand which of the methods for refining plan suggestions is preferred by the human-in-the-loop via human-subject studies.

References

Ai-Chang, M.; Bresina, J.; Charest, L.; Chase, A.; Hsu, J.-J.; Jonsson, A.; Kanefsky, B.; Morris, P.; Rajan, K.; Yglesias, J.; et al. 2004. Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems* 19(1):8–12.

Cashmore, M.; Collins, A.; Krarup, B.; Krivic, S.; Magazzeni, D.; and Smith, D. 2019. Towards explainable ai planning as a service. *arXiv preprint arXiv:1908.05059*.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proc. IJCAI*.

Chakraborti, T.; Sreedharan, S.; and Kambhampati, S. 2020. The emerging landscape of explainable ai planning and decision making. *arXiv preprint arXiv:2002.11697*.

Dragan, A. D.; Lee, K. C.; and Srinivasa, S. S. 2013. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 301–308. IEEE.

Eifler, R.; Cashmore, M.; Hoffmann, J.; Magazzeni, D.; and Steinmetz, M. 2020. A new approach to plan-space explanation: Analyzing plan-property dependencies in oversubscription planning. In *AAAI*, 9818–9826.

Ferguson, G.; Allen, J. F.; Miller, B. W.; et al. 1996. Trains-95: Towards a mixed-initiative planning assistant. In *AIPS*, 70–77.

Grover, S.; Sengupta, S.; Chakraborti, T.; Mishra, A. P.; and Kambhampati, S. 2019. ipass: A case study of the effectiveness of automated planning for decision support. *HCI Journal*.

Junker, U. 2001. Quickxplain. *Conflict Detection for Arbitatrary Constraint Propagation Algorithms Proceedings IJCAI*.

Kambhampati, S., and Talamadupula, K. 2015. Human-in-the-loop planning and decision support. *AAAI Tutorial*.

Kambhampati, S.; Knoblock, C. A.; and Yang, Q. 1995. Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning. *Artificial Intelligence*.

Kim, J.; Banks, C. J.; and Shah, J. A. 2017. Collaborative planning with encoding of users' high-level strategies. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 955–961. AAAI Press.

Krarup, B.; Cashmore, M.; Magazzeni, D.; and Miller, T. 2019. Model-based contrastive explanations for explainable planning.

Kulkarni, A.; Zha, Y.; Chakraborti, T.; Vadlamudi, S. G.; Zhang, Y.; and Kambhampati, S. 2016. Explicablility as minimizing distance from expected behavior. *arXiv preprint arXiv:1611.05497*.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. Pddl–the planning domain definition language–version 1.2. *Yale Center for Computational Vision and Control, Tech. Rep. CVC TR-98-003/DCS TR-1165*.

Mercado, J. E.; Rupp, M. A.; Chen, J. Y.; Barnes, M. J.; Barber, D.; and Procci, K. 2016. Intelligent agent transparency in human–agent teaming for multi-uxv management. *Human factors* 58(3):401–415.

Miller, T. 2018. Contrastive explanation: A structural-model approach. *arXiv preprint arXiv:1811.03163*.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*.

Mishra, A. P.; Sengupta, S.; Sreedharan, S.; Chakraborti, T.; and Kambhampati, S. 2019. Cap: A decision support system for crew scheduling using automated planning. *Naturalistic Decision Making*.

Parasuraman, R. 2000. Designing automation for human use: empirical studies and quantitative models. *Ergonomics* 43(7):931–951.

Ramírez, M., and Geffner, H. 2009. Plan recognition as planning. In *Twenty-First International Joint Conference on Artificial Intelligence*.

Sengupta, S.; Chakraborti, T.; Sreedharan, S.; Vadlamudi, S. G.; and Kambhampati, S. 2017. Radar—a proactive decision support system for human-in-the-loop planning. In *2017 AAAI Fall Symposium Series*.

Sengupta, S.; Chakraborti, T.; and Kambhampati, S. 2018. Ma-radar–a mixed-reality interface for collaborative decision making. *ICAPS UISP*.

Sheridan, T. B., and Verplank, W. L. 1978. Human and computer control of undersea teleoperators. Technical report, Massachusetts Inst of Tech Cambridge Man-Machine Systems Lab.

Smith, D. E. 2012. Planning as an iterative process. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Sohrabi, S.; Riabov, A. V.; and Udrea, O. 2016. Plan recognition as planning revisited. In *IJCAI*, 3258–3264.

Sreedharan, S.; Kambhampati, S.; et al. 2018. Handling model uncertainty and multiplicity in explanations via model reconciliation. In *ICAPS*.

Sreedharan, S.; Srivastava, S.; and Kambhampati, S. 2018. Hierarchical expertise level modeling for user specific contrastive explanations. In *IJCAI*, 4829–4836.

Talamadupula, K.; Smith, D. E.; and Kambhampati, S. 2014. The metrics matter! on the incompatibility of different flavors of replanning. *arXiv preprint arXiv:1405.2883*.

Wickens, C. D.; Li, H.; Santamaria, A.; Sebok, A.; and Sarter, N. B. 2010. Stages and levels of automation: An integrated meta-analysis. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 54, 389–393. Sage Publications Sage CA: Los Angeles, CA.

Zhang, Y.; Sreedharan, S.; Kulkarni, A.; Chakraborti, T.; Zhuo, H. H.; and Kambhampati, S. 2017. Plan explicability and predictability for robot task planning. In *2017 IEEE international conference on robotics and automation (ICRA)*, 1313–1320. IEEE.