MA–RADAR - A Mixed-Reality Interface for Collaborative Decision Making

Sailik Sengupta[*] and **Tathagata Chakraborti**[*] and **Subbarao Kambhampati**

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University

{ sailiks, tchakra2, rao } @ asu.edu

Abstract

There has been a lot of recent interest in the planning community towards adapting automated planning techniques for the role of decision support for human decision makers in the loop. A unique challenge in such settings is the presence of multiple humans collaborating during the planning process which not only requires algorithmic advances to handle issues such as diverging mental models and the establishment of common ground, but also the development of user interfaces that can facilitate the distributed decision making process among the human planners. We posit that recent advances in augmented reality technology is uniquely positioned to serve this need. For example, a mixed-reality workspace can be ideal for curating information towards the particular needs (e.g. explanations) of the individual decision makers. In this paper, we report on ongoing work along these directions and showcase MA-RADAR, the multi-agent version of the decision support system RADAR (Sengupta et al. 2017).

In (Sengupta et al. 2017), we explored the evolving roles of an automated planner in the scope of decision support for a single human in the loop. Specifically, we outlined how well-established principles of *naturalistic decision-making* and the *automation hierarchy* studied in existing literature on human-computer interaction (Parasuraman, Sheridan, and Wickens 2000; Klein 2008) can be adopted for the design of automated decision support using planners as well. In this regard, we demonstrated how the traditional role of an automated planner changes from one of plan generation to more nuanced roles of plan validation, recognition, recommendation, critique, explanations, and so on. However, most of these techniques, as well as the GUI itself, were specifically designed to deal with a single human decision maker in the loop. As we illustrate in the paper, these become ineffective in a distributed setting.

What if there are *multiple humans* in the loop?

A common feature of most collaborative planning settings is the presence of multiple human decision makers who are actively involved in the construction of the plan on a shared graphical user interface (GUI) in "control room" styled environments (Karafantis 2013; Murphy 2015). For the design of decision support technologies, this raises several unique challenges such as (1) dealing with diverse points of view, preferences, and goals; (2) diverging beliefs and mental models; (3) resolution of competing truths and establishment of common ground; and so on. Some of these issues have been highlighted recently in (Kim and Shah 2017). From the perspective of the GUI itself, multiple decision makers poses new challenges on how information is presented to the end users, not only in the way it is displayed, but also the approach to generate that information which drives the decision support infrastructure in the back-end.

What can augmented reality bring to the table?

We argue that augmented reality (AR) brings in capabilities that are uniquely suited for this purpose. This is because AR can, in effect, provide different versions of the same interface to the commanders based on their specific needs, while still preserving the convenience and efficiency of collaboration across a shared GUI. In this work, we thus will build on our previous decision support system – c.f. RADAR (Sengupta et al. 2017) – and highlight challenges, especially as it relates to the design of the interface for the decision support system, when the collaborative decision making setting is extended to deal with multiple human planners simultaneously. We will, in particular, show how –

- Augmented reality provides an effective medium of augmenting the shared GUI with private information (as studied in planning literature (Brafman and Domshlak 2008)) – thus the same plan will appear differently on the shared GUI than in the mixed-reality view where the private actions will be coupled with the public plan; and
- Augmented reality can reduce irrelevant information on the screen by porting them into the mixed-reality view. Such situations can occur, for example, when one user asks for an explanation, which the others may not require and thus should not appear on the shared GUI and potentially cause cognitive overload.

Finally, we will end with a discussion on the current work in progress and considerations of the trade-offs in AR versus distributed graphical interfaces. We note, as far as the *vitamin versus aspirin* question is concerned, AR firmly lies with the former group – after all, the humans could be equipped with separate personal screens on top of a shared

[*] Authors marked with asterix contributed equally.

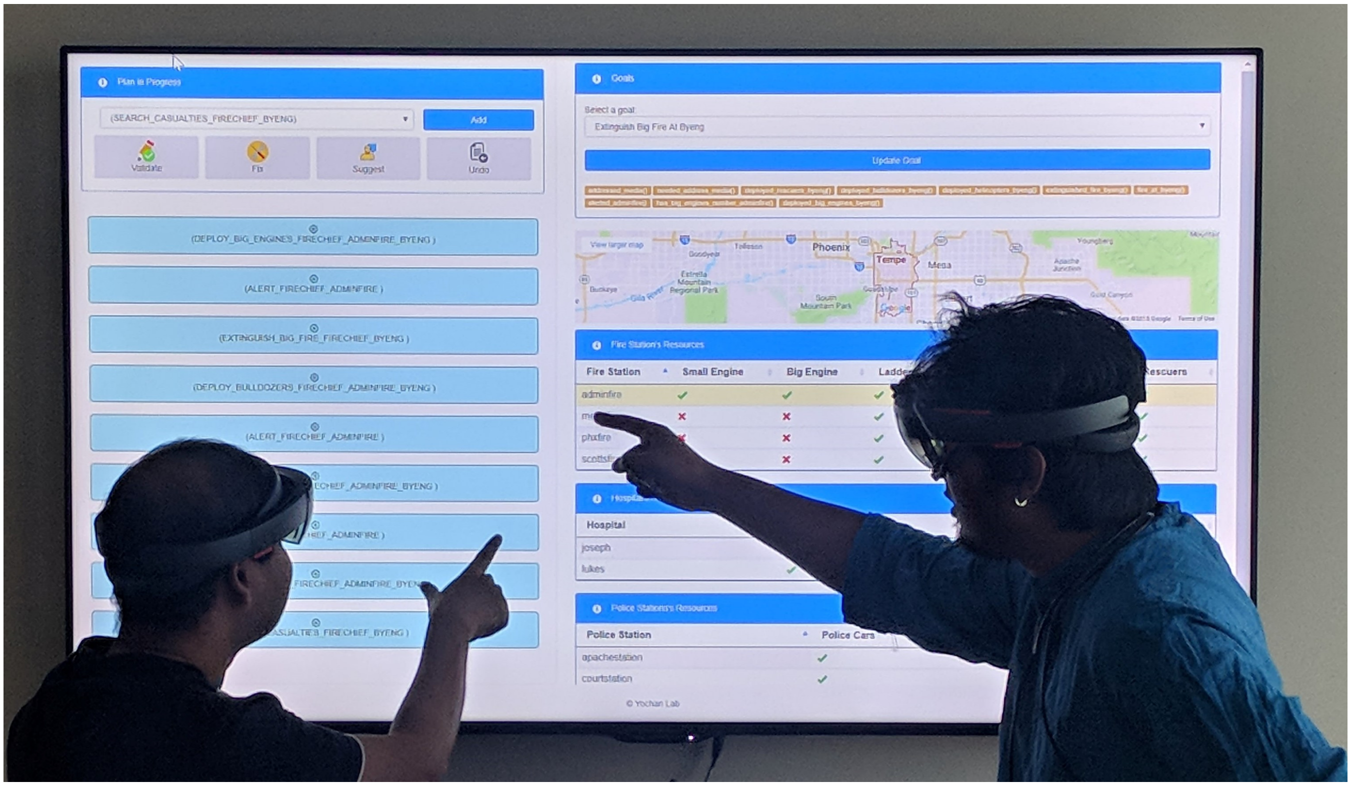

Plan in Progress
(SEARCH_CASUALTIES_FIRECHIEF_BYENG)
Add
Validate
Fix
Suggest
Undo
(DEPLOY_BIG_ENGINES_FIRECHIEF_ADMINFIRE_BYENG)
(ALERT_FIRECHIEF_ADMINFIRE)
(EXTINGUISH_BIG_FIRE_FIRECHIEF_BYENG)
(DEPLOY_BULLDOZERS_FIRECHIEF_ADMINFIRE_BYENG)
(ALERT_FIRECHIEF_ADMINFIRE)
Goals
Select a goal
Extinguish Big Fire At Byeng
Update Goal
Phoenix
Tempe
Mesa
Fire Station's Resources
Fire Station
Small Engine
Big Engine
adminfire
phxfire
Hospital
joseph
lukes
Police Station's Resources
Police Station
Police Cars
apachestation
courtstation
© Yochan Lab

Figure 1: Multiple commanders involved in the collaborative decision making process on the MA-RADAR interface. The shared interface (GUI) provides an overview of the public plan and resources, constraints, etc. pertaining to the planning problem. The mixed-reality view for each commander augments private information (such as private action in the plan) or personalized explanations of the plan for the commander. Refer to Figures 2 and 3 for the augmented view for each of these use cases.

GUI. However, we argue, and hopefully this is apparent in the demonstrations as well, that AR provides an attractive solution towards providing personalized planning interfaces to the human decision makers while still leveraging the paradigm of a shared collaborative interface of a control room accepted as the de-facto standard in these settings.

MA-RADAR

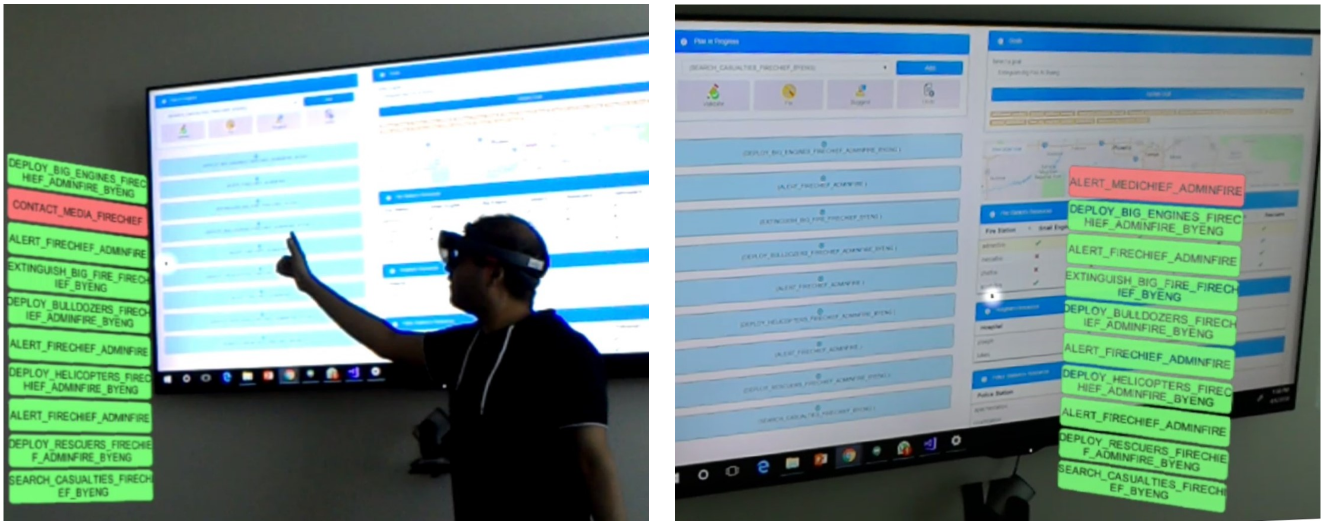
In the following, we will briefly introduce the fire fighting domain (Sengupta et al. 2017) which we will use to illustrate the UI challenges addressed in the paper.

The Fire-Fighting Domain The fire fighting domain involves extinguishing fire at a particular location (Tempe, in our case). It requires two commanders (henceforth referred to as Comm-I and Comm-II) to come up with a plan or course of action (CoA) which involves coordination with the police, medical and transport authorities. Each commander might have a personalized model of this domain, which (1) may have certain actions that are *private* to them, i.e. unknown to the other commanders; and (2) incorrect ideas about the actual domain, for example, an incorrect action definition (according to the model of the decision support agent). A detailed description of the domain used by MA-RADAR is available in (Sengupta et al. 2017). We assume that these personalized models (of the two experts) are available to MA-RADAR, which helps it to distinguish between private and public actions (Brafman and Domshlak 2008). While MA-RADAR uses a centralized model to help in validating plans and generating action or plan suggestions, the response to the users needs to be carefully curated since showing one user's private data to another user is problematic. Furthermore, explanations are inherently user specific (as are information that is being consumed by private actions only) and should not clutter a shared GUI.

Privacy Preserving Planning

In the first demonstration, we will tackle the issue of private information in the individual models of the commanders.

Background In (Brafman and Domshlak 2008) authors explored multi-agent planning scenarios where each agent has a different domain model with individual actions that can have private preconditions and effects which are not accessible to other agents. Planning in such scenarios becomes more complex because state-space search techniques have to ensure that private state variables of an agent are not exposed to other agents (Brafman 2015). As mentioned before, the interface in MA-RADAR follows this notion of private and public predicates/actions and communicates information (e.g. explanations and plan suggestions) to the user without revealing private data of another hu-


DEPLOY_BIG_ENGINES_FIRECHIEF_ADMINFIRE_BYENG
CONTACT_MEDIA_FIRECHIEF
ALERT_FIRECHIEF_ADMINFIRE
EXTINGUISH_BIG_FIRE_FIRECHIEF_BYENG
DEPLOY_BULLDOZERS_FIRECHIEF_ADMINFIRE_BYENG
ALERT_FIRECHIEF_ADMINFIRE
DEPLOY_HELICOPTERS_FIRECHIEF_ADMINFIRE_BYENG
ALERT_FIRECHIEF_ADMINFIRE
DEPLOY_RESCUERS_FIRECHIEF_ADMINFIRE_BYENG
SEARCH_CASUALTIES_FIRECHIEF_BYENG
ALERT_MEDICHIEF_ADMINFIRE
DEPLOY_BIG_ENGINES_FIRECHIEF_ADMINFIRE_BYENG
ALERT_FIRECHIEF_ADMINFIRE
EXTINGUISH_BIG_FIRE_FIRECHIEF_BYENG
DEPLOY_BULLDOZERS_FIRECHIEF_ADMINFIRE_BYENG
ALERT_FIRECHIEF_ADMINFIRE
DEPLOY_HELICOPTERS_FIRECHIEF_ADMINFIRE_BYENG
ALERT_FIRECHIEF_ADMINFIRE
DEPLOY_RESCUERS_FIRECHIEF_ADMINFIRE_BYENG
SEARCH_CASUALTIES_FIRECHIEF_BYENG

(a) Comm-I, who is responsible for communication with the media, has a private action to contact media as visible only in his POV.

(b) Comm-II, who is in charge of communicating with the medical units, has a private action to alert the medical chief in the area.

Figure 2: Mixed-reality capture illustrating how the public plan in the shared GUI can be overlayed with information on private actions (private actions are in red; public actions are in green) (Brafman and Domshlak 2008) of individual decision makers thereby still allowing the use of a shared collaborative workspace.

man in the team. Note that, from the point of view of decision support, the agent itself is not following planning algorithms as outlined in (Brafman and Domshlak 2008; Brafman 2015) since the human planners are in charge of the planning process and they, of course, are not maintaining separate priority queues in their heads. However, it might be interesting to explore how the distributed planning paradigm among humans in the presence of private information can be modeled from the perspective of decision support.

Demonstration For the purpose of our demonstration (shown in Figure 2), we assume that apart from the main task of extinguishing the fire, each of the commanders have specific tasks they need to achieve. Furthermore, only the commander (in charge of a specific task) and MA-RADAR have the knowledge of these private tasks.

In our scenario, while Comm-I is in charge of handling the communication with the media, which is an important aspect in the case of disaster response scenario, Comm-II needs to take care of all communication and deployment of medical help for rescued victims. The private actions of the two commanders follow. To reduce clutter, we only provide the action names as opposed to the full action definition (which also have the private predicates).

> Comm-I
> CONTACT_MEDIA
> ADDRESS_MEDIA
>
> Comm-II
> ALERT_MEDICHIE
> ATTEND_CASUALTIES
> ISSUE_LOCAL_ALERT
> SET_UP_HELPLINE

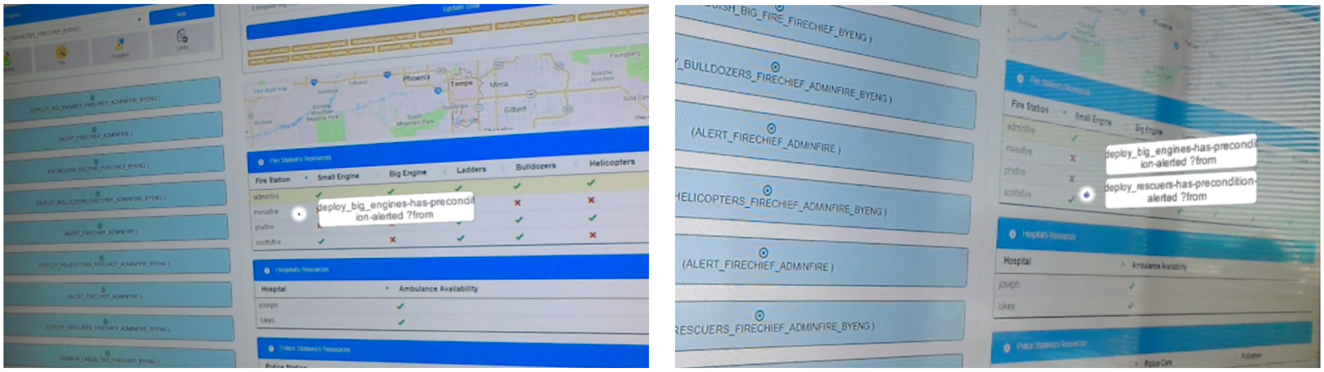
When the commanders ask MA-RADAR to suggest missing actions or complete the plan in order to achieve the goal of extinguishing the (big) fire, it needs to communicate most of the private actions mentioned here, but only to the specific commander in charge of the private task. Showing these on the common user interface would result in (1) confusion as each commander is oblivious to the private actions of the other commander and (2) loss of privacy which might be important in complex decision making scenarios with multiple commanders (e.g. army, navy, etc.).

Thus, in such scenarios, MA-RADAR tries to display these private actions in the augmented view of each commander (highlighted in red in Figure 2). Since each action in the plan occupies a substantial amount of space in the 3D-view, we show only 10 actions at any point in time. This ensures that the commander does not have to stare away from the common interface, which can lead to loss of situational awareness as other commanders might make changes to common elements in that time (e.g. by updating resources, rearranging or removing public actions, etc.).

Multi-Model Explanations

The second demonstration looks at *plan explanations for model reconciliation* introduced in (Chakraborti et al. 2017) – the aim of explanations of this form is to provide updates to the user's possibly faulty understanding of the planning problem to make sure that the optimal plans in the planner's model are also optimal in the human's. Thus the process of model reconciliation is crucial in maintaining that the decision support agent is on the same page as the human in the loop and thus the establishment of common ground.

Background The above model reconciliation process is only feasible if inconsistencies of the planner's model with


deploy_big_engines-has-precondition-alerted ?from
deploy_big_engines-has-precondition-alerted ?from
deploy_rescuers-has-precondition-alerted ?from
(ALERT_FIRECHIEF_ADMINFIRE)
HELICOPTERS_FIRECHIEF_ADMINFIRE_BYENG)
RESCUERS_FIRECHIEF_ADMINFIRE_BYENG)

(a) Comm-I, who is unaware of the procedure that a fire-chief needs to be alerted first before deploying the fire engines, is provided this explanation to justify the suggested (public part of the) plan.

(b) Comm-II, unaware that a fire-chief needs to be alerted before deploying any kind of resources (fire engines or rescuers) from a fire station, is provided both of these model updates as explanations.

Figure 3: Mixed-reality capture illustrating how the multi-model explanation generation algorithm (Sreedharan, Chakraborti, and Kambhampati 2018) can be used to provide targeted explanations to each commander based on their models without inundating the other commanders with superfluous or unsolicited information.

the user's mental model are known precisely, or in general, if there is a single model that needs to be reconciled. Instead, in a team decision making setting, the decision support agent may end up having to explain its decisions with respect to a *set of models* one for each human in the loop. In this situation, MA−RADAR can look to compute explanations for each possible configuration. However, computing separate explanations (Chakraborti et al. 2017) for each agent can result in situations where the explanations computed for individual models independently are not consistent across all the possible target domains. In the case of multiple teammates being explained to, this may cause confusion and loss of trust, especially in teaming with humans who are known to rely on shared mental models. Instead, in (Sreedharan, Chakraborti, and Kambhampati 2018) we proposed an explanation generation process such that a single model update that makes the given plan optimal (and hence explained) in all the updated domains (or in all possible domains).

In order to deal with multiple humans in the loop, we have thus shifted towards (Sreedharan, Chakraborti, and Kambhampati 2018) instead of (Chakraborti et al. 2017) as originally demonstrated in (Sengupta et al. 2017). However, from the point of view of the interface, there still remains the matter of filtering out superfluous information (due to the single explanation or model update being computed that suffice for all the models) as they are being presented to the individual users. We will illustrate this next.

Demonstration For our demonstration (shown in Figure 3), we assume that the two commanders have different understanding about the domain used by MA−RADAR. We further assume that this knowledge about the actual domain is (1) different for the different commanders, which is often the case in real-world scenarios (as there may be many ways of being incorrect about the correct procedure) and (2) these explanations are, for the purpose of this example, limited to updates about public actions. In scenarios where explanations are about private predicates or private actions of a commander, MA−RADAR , with the models of both the commanders, filters out these (private explanations) when generating explanations for the other commander.

In order to highlight the domain differences, we will first show the part of the actual model (that MA−RADAR has) about which the commanders have incorrect idea.

```
(:action deploy_big_engines
  :parameters (?a – fire ?from –
     firestation ?to – pois)
  :precondition (and
    (alerted ?from)
    (has_big_engines_number ?from)
  )
  :effect (and
    (not (alerted ?from))
    ...
  )
)
(:action deploy_rescuers
  :parameters (?a – fire ?from –
     firestation ?to – pois)
  :precondition (and
    (alerted ?from)
    (has_rescuers_number ?from)
  )
  :effect (and
    ...
  )
)
```

We now show the model that Comm-I has, where the precondition for alerting the authority at a fire-station is missing as a precondition for deploying (big) fire engines –

```
(:action deploy_big_engines
  :parameters (?a – fire ?from –
     firestation ?to – pois)
  :precondition (and
     (has_big_engines_number ?from)
     )
  :effect (and
    (not (alerted ?from))
    ...
  )
)
(:action deploy_rescuers
  :parameters (?a - fire ?from -
     firestation ?to - pois)
  :precondition  (and
    (alerted ?from)
    (has_rescuers_number ?from)
  )
  :effect (and
    ...
  )
)
```

For Comm-II, who is completely unaware that fire-stations need to be alerted in order to deploy fire engines or rescuers, the domain model looks as follows –

```
(:action deploy_big_engines
  :parameters (?a - fire ?from -
     firestation ?to - pois)
  :precondition (and
     (has_big_engines_number ?from)
     )
  :effect (and
    (not (alerted ?from))
    ...
  )
)
(:action deploy_rescuers
  :parameters (?a - fire ?from -
     firestation ?to - pois)
  :precondition (and
     (has_rescuers_number ?from)
     )
  :effect (and
    ...
  )
)
```

When the commanders ask MA-RADAR to suggest a plan (or complete a plan) in order to achieve the goal of extinguishing big fire, it will suggest a plan that has both the actions of deploying big engines and rescuers. Since both of these actions need to alert the authority at the fire station, there will be two alert_firechief actions which makes the alerted_firechief proposition (which is a precondition of these two actions in the original domain) true.

In this situation, although both the commanders might be surprised at the suggested plan and ask for explanations, Comm-I just needs to be told about the missing precondition of the deploy_big_fire_engine action, whereas, Comm-II, in addition to that explanation, also needs to be told about the missing precondition of the action deploy_rescuers. The augmented reality workspace helps us to provide personalized explanations to both the commanders (see Figure 3).

Work in Progress

Currently, we are working on making the mixed-reality display more interactive and porting more of the utilities in the shared GUI into it. This, of course, raises interesting challenges from the point of view of intra-team interactions –

- "Hiding" much of the interface, even though not relevant to the team, can cause inefficiency and friction in the collaborative process. It may well be possible that revealing too little information as needed can cause lack of situational awareness while leaving it all out there is likely to cause cognitive overload. As such, there needs to be a delicate balance between how much of the shared GUI can be abstracted out into the mixed reality workspace.
- Allowing for a distributed workspace also requires processing of concurrent requests (for replanning, validation, etc.) which needs to be handled gracefully at the frontend – e.g. two commanders making concurrent edits on the public plan in their own mixed-reality spaces is undesirable and needs to be orchestrated effectively.

We hope to discuss some of these challenges, as well as report on our work in progress, at the workshop.

References

Brafman, R. I., and Domshlak, C. 2008. From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*, 28–35.

Brafman, R. I. 2015. A privacy preserving algorithm for multi-agent planning and search. In *IJCAI*, 1530–1536.

Chakraborti, T.; Sreedharan, S.; Kambhampati, S.; and Zhang, Y. 2017. Explanation generation as model reconciliation in multi-model planning. Technical report.

Karafantis, L. 2013. NASA's Control Centers: Design and History. Engineering and Technology.

Kim, J., and Shah, J. 2017. Towards intelligent decision support in human team planning. In *AAAI Fall Symposium on Human-Agent Groups*.

Klein, G. 2008. Naturalistic decision making. *The Journal of the Human Factors and Ergonomics Society*.

Murphy, M. 2015. Searching for Eureka: IBMs path back to greatness, and how it could change the world. Quartz.

Parasuraman, R.; Sheridan, T. B.; and Wickens, C. D. 2000. A model for types and levels of human interaction with automation. *Trans. Sys. Man Cyber. Part A*.

Sengupta, S.; Chakraborti, T.; Sreedharan; and Kambhampati, S. 2017. RADAR - A Proactive Decision Support System for Human-in-the-Loop Planning. In *ICAPS Workshop on User Interfaces for Scheduling and Planning (UISP)*.

Sreedharan, S.; Chakraborti, T.; and Kambhampati, S. 2018. Handling Model Uncertainty and Multiplicity in Explanations as Model Reconciliation. In *ICAPS*.