# HELP! Providing Proactive Support in the Presence of Knowledge Asymmetrys

Turgay Caglar
Colorado State University
Fort Collins, United States
tcaglar@colostaste.edu

Sarath Sreedharan
Colorado State University
Fort Collins, United States

## ABSTRACT

While the development of proactive personal assistants has been a popular topic within AI research, most research in this direction tends to focus on a small subset of possible interaction settings. An important setting that is often overlooked is one where the users may have an incomplete or incorrect understanding of the task. This could lead to the user following incorrect plans with potentially disastrous consequences. Supporting such settings requires agents that are able to detect when the user's actions might be leading them to a possibly undesirable state and, if they are, intervene so the user can correct their course of actions. For the former problem, we introduce a novel planning compilation that transforms the task of estimating the likelihood of task failures into a probabilistic goal recognition problem. This allows us to leverage the existing goal recognition techniques to detect the likelihood of failure. For the intervention problem, we use model search algorithms to detect the set of minimal model updates that could help users identify valid plans. These identified model updates become the basis for agent intervention. We further extend the proposed approach by developing methods for pre-emptive interventions, to prevent the users from performing actions that might result in eventual plan failure. We show how we can identify such intervention points by using an efficient approximation of the true intervention problems, which are best represented as a Partially Observable Markov Decision-Process (POMDP). To substantiate our claims and demonstrate the applicability of our methodology, we have conducted exhaustive evaluations across a diverse range of planning benchmarks. These tests have consistently shown the robustness and adaptability of our approach, further solidifying its potential utility in real-world applications.

## KEYWORDS

Proactive assistance; Pre-emptive intervention; Task failure recognition; Knowledge asymmetry

**ACM Reference Format:**
Turgay Caglar and Sarath Sreedharan. 2024. HELP! Providing Proactive Support in the Presence of Knowledge Asymmetrys. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 10 pages.



*Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.* © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

## 1 INTRODUCTION

There is a long history within AI for developing proactive personal assistants [5, 20, 21, 29, 31, 47, 54, 57]. These are automated agents that are meant to keep track of the activities of a user and their eventual goals and offer help whenever the agent determines that the user may benefit from it. While many previous works have looked at such support problems, most of the works that propose formal support models seem to focus on a few specific settings. In particular, there is a lot of focus on what might be considered 'serendipitous support,' where the user tries to achieve their goal, and the agent tries to reduce the burden placed on the user (cf. [7, 18, 27, 50, 58]). Works have also looked at supporting users with cognitive disabilities [4, 30, 52, 53].

However, a direction of work that has gotten relatively less attention is the use of disembodied assistants in the presence of knowledge assymmetry i.e., support users when their estimate about the task may have changed or differs from the agent's true estimate. As such, the plans users devise may not be valid. The potential use cases here could range from safety-critical, imagine a system alerting a rescue worker in a disaster scenario about a collapsed wall blocking their path, to the quotidian, a system that informs the user about heavy traffic in their normal route to work. Also, the setting simplifies the kinds of intervention required from the agent's end to merely informing the user about task information they might not have been aware of previously.

The primary challenge with these works is identifying the degree to which (if at all) the difference in task knowledge impacts the user's ability to generate valid plans. If they do, the agent should be able to recognize what information about the underlying model should be provided to the user so they can choose a new valid plan. As with other works in this direction, we will assume that the user's actual plan is not known upfront to the assistive agent. Additionally, the difference in the user's estimate of the task and the agent's might be significant. Depending on the plan the user selects, only a subset of differences between the agent's estimate and the user's own estimate might be relevant, where relevancy is determined by whether it impacts the plan's validity. As such, we will start by proposing a way to estimate the probability that a user may follow an invalid plan in the agent's task model. We will introduce a novel planning compilation that will map the problem of detecting failure into that of probabilistic goal recognition and then employ existing goal recognition tools to estimate the probabilities.

Additionally, for a given state, we also show how one could adapt model space search [10], to generate the minimal set of information about model changes that need to be passed to the user so they are guaranteed not to follow a potentially incorrect plan. One could see

such information being provided to the user once the confidence the agent has in the user making a mistake crosses a certain threshold.

However, as we will see, relying purely on the probability of failure has downsides. One would want the assistive agent to make suggestions or intervene early enough that the user can take corrective actions and avoid potential mistakes. For example, in the case of heavy traffic, you would expect the agent to notify the user before they enter the road and not after they are stuck in the traffic (even though the agent can be very confident that the user is following a bad plan in the latter case). In mission-critical non-ergodic domains, such early intervention can be quite critical. As we will see in Section 6, while the true problem of generating pre-emptive agent intervention may be best expressed as a Partially Observable Markov decision process (POMDP), we can approximate the problem that is guaranteed to generate a more cautious policy.

To summarize, the main contributions of the paper are as follows:

- We develop a novel planning compilation that maps the problem of estimating failure probability to a goal recognition problem. (Section 4)
- We update the model search algorithm to find the minimal set of model updates that need to be provided to the user to avoid potential failures. (Section 5)
- We develop a decision-theoretic method to identify scenarios that require early intervention. (Section 6)
- We evaluate the various proposed solutions on several planning benchmarks. (Section 8)

## 2 BACKGROUND

This paper will focus on problems that can be captured using factored deterministic goal-directed planning models [16, 46]. Traditionally, such models are captured by using a tuple of the form $\mathcal{M} = \langle F, A, I, G \rangle$, where $F$ is a set of propositional fluents that is used to define the state space over which the model is defined (each state $s$ is uniquely represented by the set of fluents that is true in the state $s \subseteq F$), $A$ is the set of actions that can be executed, $I \subseteq F$ the initial state, $G \subseteq F$ is the goal description. Each action $a \in A$ is further defined by the tuple $a = \langle pre(a), ceff(a), add(a), del(a) \rangle$, where $pre(a)$ is a conjunctive propositional formula defined over $F$, $ceff(a)$ are the set of conditional effects and $add(a) \subseteq F$, $del(a) \subseteq F$ the unconditioned add and delete effects associated with the action. Conditional effects are only applied when the state meets a set of conditions, and unconditioned add and delete effects are always applied when the action is executable. Each conditional effect $e \in ceff(a)$, can be further defined using a tuple of the form $\langle C(e), add(e), del(e) \rangle$. Here $C(e)$, a propositional logical formula, represents the conditions under which the specific effect is applied as part of action execution. The components $add(e)$ and $del(e)$ are the add and delete effects applied when the overall conditional effect is applied. For a given planning model, we can define a transition function $\gamma_{\mathcal{M}} : 2^F \rightarrow 2^F$ as follows:

$$\gamma_{\mathcal{M}}(s) = \begin{cases} (s \setminus \text{del\_set}(s,a)) \cup \text{add\_set}(s,a), & \text{if } s \models pre \\ undefined & \text{otherwise} \end{cases}$$

Where,

$$\text{add\_set}(s,a) = add(a) \bigcup_{\langle C(e), add(e), del(e) \rangle \text{ and } C(e) \models s} add(e)$$

and

$$\text{del\_set}(s,a) = del(a) \bigcup_{\langle C(e), add(e), del(e) \rangle \text{ and } C(e) \models s} del(e)$$

Note that when we use a state $s$ in the context of the entailment operator, we are effectively considering the logical formula obtained by considering the conjunction of all positive literals corresponding to fluents that are part of that state and the negative literals corresponding to the fluents that are absent from that state. We will also overload the transition function to apply to action sequences as well. For models where the conditional effect set is empty, we will simplify the entire action representation and represent it using the tuple $\langle pre(a), add(a), del(a) \rangle$. We will generally consider settings where actions have unit costs. However, the proposed approach can easily be extended to cases where actions have differing action costs. A solution to a planning problem is a plan, where a plan $\pi = \langle a_1, ..., a_k \rangle$, is simply an action sequence that satisfies the requirement $\gamma_{\mathcal{M}}(\pi, I) \models G$. We will refer to the plan with the lowest cost (in this case, being equivalent to the shortest) as the optimal plan. Additionally, we will sometime refer to the tuple consisting of just the fluent and action set (i.e., $\langle F, A \rangle$ as the domain of the planning problem.

This paper also involves finding model updates to generate planning models with the required properties. To do this, we will follow the model-space search paradigm followed by methods like model reconciliation [10, 42, 43], and similar to these earlier works, we will rely on a model parameterization function to represent each possible model by a set of propositional factors. Since we will be using these parameterization methods over the original models, we will define a parameterization function that only supports model types without conditional effects.

Specifically, we will follow the conventions set by Sreedharan et al. [42] and define a model parameterization function $\Gamma$ for a set of fluents $F$ and action labels $A$, which is defined over a set of model parameters $\mathcal{F}^{(F,A)}$, where

$$\begin{aligned} \mathcal{F}^{(F,A)} = \; & \{\textit{init-has-f} \mid f \in F\} \cup \{\textit{goal-has-f} \mid f \in F\} \cup \\ & \bigcup_{a \in A} \{\textit{a-has-pos-prec-f}, \textit{a-has-neg-prec-f}, \\ & \quad \textit{a-has-add-f}, \textit{a-has-del-f} \mid f \in F\}. \end{aligned}$$

And the model parameterization function itself is defined as $\Gamma(\mathcal{M})$ , which is a mapping to a state $s \subseteq \mathcal{F}$, is defined by

$$\begin{aligned} \tau_I &= \{\textit{init-has-f} \mid f \in I\} \\ \tau_G &= \{\textit{goal-has-g} \mid g \in G\} \\ \tau_{pre_+(a)} &= \{\textit{a-has-pos-prec-f} \mid f \in pre_+(a)\} \\ \tau_{pre_-(a)} &= \{\textit{a-has-neg-prec-f} \mid f \in pre_-(a)\} \\ \tau_{add(a)} &= \{\textit{a-has-add-f} \mid f \in add(a)\} \\ \tau_{del(a)} &= \{\textit{a-has-del-f} \mid f \in del(a)\} \\ \tau_a &= \tau_{pre_+(a)} \cup \tau_{pre_-(a)} \cup \tau_{add(a)} \cup \tau_{del(a)} \\ \tau_A &= \bigcup_{a \in A^{\mathcal{M}}} \tau_a \\ \Gamma(\mathcal{M}) &= \tau_I \cup \tau_G \cup \tau_A \end{aligned}$$

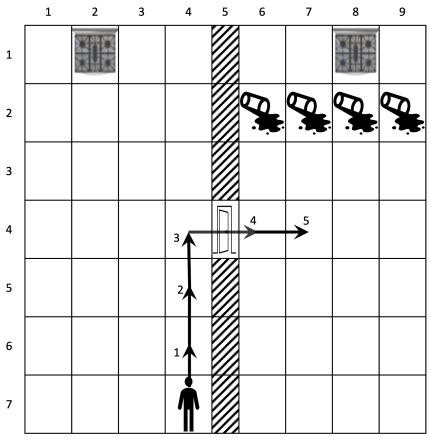


**Figure 1: A graphical representation of our running example involving an AI agent observing a human operating in a kitchen.**

Where $pre_+(a)$ is the positive literals that are part of the precondition for $a$ and $pre\_(a)$ the negative ones.

In our work, we map the problem of proactive intervention to that of solving a Partially Observable Markov Decision Process (POMDP) [6, 26, 34]. Generally, POMDPs can be described with 8-tuples $\langle S, A, O, T, C, \Omega, \mu_0, \delta \rangle$, where $S$, $A$, and $O$ represent the agent's state, action, and observation space, respectively. In each step, the agent takes an action $a \in A$ at a state $s$. This results in the agent receiving an observation $o \in O$ and the environment state transitioning to a state $s'$. The transition probabilities for the state are captured by $T$, which is a set of conditional probabilities of the form $T(s'|s, a)$. $C$ is a cost function that maps all state and action pairs to a cost. Similarly, $\Omega$ captures observation probabilities $\Omega(o|s', a)$. $\mu_0$ is the initial distribution over the states and finally, $\delta \in [0, 1)$ is the discount factor. The goal here would be to generate behavior that minimizes the expected discounted total cost. Since the agent doesn't know the exact environment state, it can track its current state estimate by tracking the whole history of observations received or converting them into a posterior distribution over the likelihood of possible states (starting with $\mu_0$ as the prior beliefs). We will refer to the latter representation as a belief state and use $\mathbb{B}$ to represent the set of all belief states. A policy for a POMDP can take the form of a function that maps belief states to actions. In this case, the value function ($V^\pi : \mathbb{B} \to \mathbb{R}$) for a policy $\pi$ returns the expected total discounted cost received by executing a policy at a belief state, and the Q value function ($Q^\pi : \mathbb{B} \times A \to \mathbb{R}$) returns the expected total discounted cost received by executing an action at a belief state and then following the policy. An optimal policy is the one with the lowest expected total discounted cost, and the corresponding value and Q value functions are represented as $V^*$ and $Q^*$.

## 3 RUNNING EXAMPLE

Imagine an AI agent observing a human in a kitchen setup. The agent is tasked with the responsibility of ensuring the safety of the human. This robot monitors the human's actions, gauges task failure risks, and steps in to prevent errors. Figure 1 depicts our "Kitchen Domain" example, showing a human navigating a specific 9x7 grid kitchen layout. The human starts at point (4,7). The human can move diagonally, vertically, or horizontally through the kitchen, each move being a unit cost action. A wall, marked by diagonal arrows along the y-axis, divides the kitchen. A one-way door at (5,4) allows movement only from the kitchen's left to the right side. The human's objective is to switch on one of the ovens at (2,1) or (8,1). However, unknown to the human, there are oil spills blocking paths to the right oven, and stepping on the spill could potentially lead to the human injuring themselves. The human ignorance about the oil spill may come from the fact that the spill just happened and wasn't present the last time the human visited the room. On the other hand, the assistive agent uses information received from sophisticated sensors placed all around the kitchen to generate an accurate estimate of the overall state. Arrows in Figure 1 indicate a set of possible observations the agent could have received at various points in time. The objective of this paper is to build an approach that can identify when the human may be headed to a possibly unsafe situation and intervene before they make a mistake.

To ground this example within the framework of our problem, we can see that this example corresponds to a scenario with knowledge asymmetry. Here, the human is unaware of the oil spills, while the AI agent knows about its locations. As such, if we encode the knowledge of the human and the agent into planning models, we would get two distinct models that allow for different valid plans. Particularly in the human model, there are additional plans that involve the human going to the stove on the right.

The first goal of the agent would be to use the observations received from the human activity to determine whether the human is currently executing a plan that involves moving through one of the cells with oil in it. Given the exponential blow-up of plan space for factored planning settings, it would be infeasible to iterate over all potential human plans that could fail explicitly and calculate the probability of the human following one of these invalid plans. As such, we need to develop methods that can approximate this likelihood without the need to enumerate all possible plans.

Once the system has built enough confidence that the human may in fact be heading to an unsafe state, they need to intervene. In this case, the reason why the human is engaging in this unsafe behavior is the underlying knowledge asymmetry; as such, one way to intervene would be to help resolve this asymmetry. Specifically, the agent could inform the user about the oil spill so they don't move into those cells. However, it is worth noting that the agent should only inform the user about the oil spills if it has high confidence the human is headed towards it. If the human received this information when they had no intention of heading to the room on the right, the agent would just be introducing cognitive load on the human's end for no reason (which could potentially lead to the human not trusting the system and potentially ignoring it in the future). Such considerations become even more important in cases where the difference between the human model and the robot model could be quite significant. In such cases, it also becomes important that the agent should be able to recognize what is the minimal set of information that it can give to the human to avoid potential mistakes. Dumping all the model differences onto the

human would again result in the human getting overwhelmed and even potentially ignoring important information.

The question of when to provide the human with the information could be as important as what information to provide. One possibility might be to wait until the agent's confidence crosses a certain threshold or, at the very least, the agent is more confident about the human making a mistake than not making one. However, in the example laid out here, that point is reached once the human crosses through the door. While intervening after that point could help prevent the human from stepping on the oil, it would leave the human trapped in the right room and unable to complete their task. As such, at every time step, the system needs to reason about the possible cost of giving model information when unnecessary (there by incurring some penalty associated with adding cognitive load at the human's end) and the possibility that the user might take an action that would prevent them from ever reaching the goal. To do this form of reasoning, the agent not only needs to consider the probability that the human is following an invalid plan but also consider what exact next steps the user could follow (along with their likelihood). The user would need to use these probabilities with the costs associated with each outcome to determine the right course of action. In this case, this would correspond to the agent informing the human about the oil spill, as soon as the human reaches the door.

## 4 MODEL COMPILATION FOR FAILURE DETECTION

We will consider a basic setting where a human operates in an environment. The human maintains some beliefs about the task, which can be represented mathematically via a model $\mathcal{M}^H = \langle F, A^H, I^H, G^H \rangle$. Now, we have an assistive agent tracking human actions and trying to evaluate the likelihood of success. The agent maintains its estimate of the task that we will denote as $\mathcal{M}^\alpha = \langle F, A^\alpha, I^\alpha, G^\alpha \rangle$. We will refer to this pair of models $\mathbb{M}^\alpha = \langle \mathcal{M}^H, \mathcal{M}^\alpha \rangle$, as the *assistive model pair*. To simplify the discussions, we will assume that the action definitions in both models have empty conditions effect sets and share the same set of action labels. Additionally, we will assume that the $\mathcal{M}^\alpha$ is a more accurate task representation than $\mathcal{M}^H$ and that they might differ over any of the model components. For the running example, the difference between the two models is primarily in the initial state, and the initial state in the human model is missing propositions related to the oil spills next to the right oven.

At a given timestep $t$, let $O_t$ be the series of actions that the human has performed, now the first step would be to estimate the likelihood that the plan being pursued by the human (i.e., $\pi_H$) will fail. More formally, we are trying to estimate

Definition 1. *For a given assistive model pair, $\mathbb{M}^\alpha = \langle \mathcal{M}^H, \mathcal{M}^\alpha \rangle$, the* likelihood of failure *given an observation sequence $O_t$, or $\mathcal{P}_F(O_t, \mathbb{M}^\alpha)$, is equal to the probability $\sum_{\pi \in \Pi_F^H} P_H(\pi | O_t, \mathcal{M}^H)$, where $P_H$ gives the probability of the human selecting a plan given their current understanding of the task, and $\Pi_F^H$ gives the set of plans that succeeds in the human model but fails in the agent model, i.e., $\Pi_F^H = \{\pi | \gamma_{\mathcal{M}^H}(I^H, \pi) \models G^H, \gamma_{\mathcal{M}^\alpha}(I^\alpha, \pi) \not\models G^R\}$.*

*Likelihood of failure*, thus captures the marginal probability of the human selecting a plan they think will succeed but fails as per the agent's environment model.

Now to calculate this probability, we will employ a compiled model that combines both the human model estimate and the agent one. The basic intuition is that we want to build a model that supports generating all valid plans in the human model but can also track their status in the agent model.

More formally, we will represent this model as $\mathcal{M}^C = \langle F^C, A^C, I^C, G^C \rangle$. Here the new fluent $F^C$ set consists of all the original fluents, a copy for each fluent that will be used to track the agent state, and finally, a proposition called `plan_fail` to detect plan failure, i.e.,

$$F^C = F \cup \alpha(F) \cup \{\texttt{plan\_fail}\}$$

Where $\alpha(F)$ are the copies of the fluent made for the agent. This means that the compiled model will contain two copies for each original propositional fluent, and we will use $\alpha()$ as a function to map the original fluent to the agent copy. For example, in our running example, the compiled model will have two `stove_on` propositions, the original one and a new proposition $\alpha$(`stove_on`).

Coming now to the actions $A^C$, we create a copy for each human action with the same preconditions and effects but now include two new sets of conditional effects, one that corresponds to a case where the action succeeds in the agent model too (and isn't following some previous action that may have failed in the agent model) and one that corresponds to the failure in the agent model. More specifically, for action $a^H \in A^H$ (with a corresponding action $a^\alpha$ in the agent model), we will have a corresponding action $a^C = \langle pre(a^C), ceff(a^C), add(a^C), del(a^C) \rangle$, such that

$$pre(a^C) = pre(a^H),\ add(a^C) = add(a^H), \text{and } add(a^C) = del(a^H)$$

Now for the conditional effects, we have

$$\begin{aligned} ceff(a^C) = \{&\langle \alpha(pre(a^\alpha)) \wedge \neg\texttt{plan\_fail}, \\ &\alpha(add(a^\alpha)), \alpha(del(a^\alpha)) \rangle, \\ &\langle \neg\alpha(pre(a^\alpha)), \{\texttt{plan\_fail}\}, \{\} \rangle\} \end{aligned}$$

The initial state here consists of the original human initial state and then the copy of the agent's estimate of the initial state

$$I^C = I^H \cup \alpha(I^\alpha)$$

Finally, coming to the goal, the first goal we will consider if one where we are trying to find a plan where the human goal is met and the proposition `plan_fail` is also satisfied.

$$G^C = G^H \cup \{\texttt{plan\_fail}\}$$

A plan that will satisfy this goal would be one that will work on the human model but not on the agent one. This brings us to the first proposition, namely that the failure set ($\Pi_F^H$) only consists of plans that satisfy this goal

Proposition 1. *For a plan $\pi$ is part of $\Pi_F^H$, if and only if, $\gamma(I^H, \Pi^C) \models G^C$.*

The proof for this proposition is relatively straightforward. Given the structure of the preconditions and the the add effects, any plan that satisfies the human goal will still result in a state where the part of the state defined with $F$ will satisfy the human goal specification.

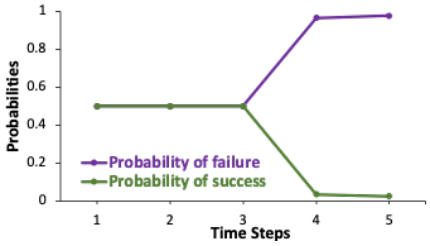


**Figure 2: Probabilities for goal failure, and not goal failure, as derived for our running example.**

Additionally, if the plan wasn't valid in the agent model, there must be at least one action in the plan where the failure conditional effect will be executed and thus producing `plan_fail`. The 'only if' part can be proved using a similar reasoning line.

Now to show that we can use the compiled model to calculate the probability, we need to show that the distribution of plans follows the original distribution in the human model. This requires us to adopt a model of decision-making for the human. A natural choice is the noisy-rational model [17, 45], which has been widely used to model human-AI interaction. Under this model, the likelihood of the human selecting an action sequence is given as

$$P_H(\pi|\mathcal{M}^H) \propto e^{-1\times C_{\mathcal{M}^H}(\pi)}$$

Where $C_{\mathcal{M}^H}(\pi) = |\pi|$ if the action sequence is valid (hence a plan) in $\mathcal{M}^H$, else it is equal to $\infty$. Now if we created a new model $\mathcal{M}^{C'} = \langle F^C, A^C, I^C, G^H\rangle$, we can see that the distribution plans under decision-making model will match the original distribution for $\mathcal{M}^H$.

Proposition 2. *For noisy rational decision-making models, we can see that*

$$P_H(\pi|\mathcal{M}^H) = P_H(\pi|\mathcal{M}^{C'})$$

*for every action sequence $\pi$.*

This follows from the fact that every action sequence that is valid in $\mathcal{M}^H$ is valid in $\mathcal{M}^{C'}$) and vice-versa. Similarly, any action sequence invalid in $\mathcal{M}^H$ is invalid $\mathcal{M}^{C'}$) and vice-versa. Thus the normalization is done over the same set, and since the cost of valid plans is conserved across the two models, the probabilities stay the same.

Given these two propositions, we can assert that the probability of failure is the probability that the human follows a plan that satisfies the goal $G^H \cup \{\texttt{plan\_fail}\}$, i.e., $G^C$ given an observation $O_t$, more formally,

$$\mathcal{P}_F(O_t, \mathbb{M}^\alpha) = \sum_{\pi, \gamma(\pi, I^C) \models G^C} P(\pi|O_t, \mathcal{M}^{C'})$$

Which in turn can be formulated as a probabilistic goal recognition problem between two mutually exclusive goals $G^H \cup \{\texttt{plan\_fail}\}$ and $G^H \cup \{\neg\texttt{plan\_fail}\}$, for the planning domain $\langle F^C, A^C\rangle$, initial state $I^C$ and observation sequence $O^C$. For this purpose, we can direct use methods like the one proposed by Ramírez and Geffner [36], which implicitly uses a noisy rational model. Figure 2 illustrates the probabilities associated with task failure and task success for each observed time slice in the running example.

## 5 MINIMAL INTERVENTIONAL INFORMATION

As discussed in the previous sections, the source of human confusion is their misunderstanding regarding the task. Additionally, the agent has access to a more accurate representation of the task. As such, a way to avoid human mistakes might be by informing them about potential differences in tasks. However, informing them about all the differences might be overwhelming and unnecessary. For example, in the running example, knowing the fact that the weather outside the house has changed will not deter the user from going down the wrong path. The goal thus becomes to find the minimal set of model updates that need to be made to the human model, per the agent model, that will ensure that in the resulting model, the human can't select a plan that will result in failure. Reader's familiar with model reconciliation [9, 10, 43, 44] literature will note the similarity of the description with that of MCE explanations. However, unlike MCE explanations, where the goal is to identify a set of model updates that will ensure that a specific plan is optimal in the updated model, our objective is to ensure that the probability of the human selecting a failing plan is zero, or more formally

Definition 2. *For a given assistive model pair, $\mathbb{M}^\alpha = \langle \mathcal{M}^H, \mathcal{M}^\alpha\rangle$ and an observation sequence $O_t$, the Minimal Intervention Information or MII is given by a pair of model updates of the form $\mathcal{E} = (\mathcal{E}^+, \mathcal{E}^-)$, such that*

- C1 $\mathcal{E}^+ \subseteq (\Gamma(\mathcal{M}^\alpha) \setminus \Gamma(\mathcal{M}^H))$ *and* $\mathcal{E}^- \subseteq (\Gamma(\mathcal{M}^H) \setminus \Gamma(\mathcal{M}^\alpha))$
- C2 *The probability of failure for the updated model pair $\tilde{\mathbb{M}}^\alpha = \langle \tilde{\mathcal{M}}^H, \mathcal{M}^\alpha\rangle$ is zero, where $\tilde{\mathcal{M}}^H = \Gamma^{-1}(\tilde{\mathcal{M}}^H \setminus \mathcal{E}^-) \cup \mathcal{E}^+$*
- C3 *There exists no pair $\widetilde{\mathcal{E}}$ that satisfies C1 and C2, such that $|\widetilde{\mathcal{E}}^+| + |\widetilde{\mathcal{E}}^-| < |\mathcal{E}^+| + |\mathcal{E}^-|$*

The model updates which satisfy C1 and C2, but not C3 (hence aren't minimal) will be referred to as simply valid interventional information (or VII)

Following the discussion in the previous section, this is equivalent to finding the minimal set of model updates to be applied to the human model so that the corresponding compiled model is unsolvable when the initial state is set to the one obtained by applying the observations (we will only consider observation sequences that are valid in both human and agent model).

Now we can identify such MII model updates by selecting any existing model-space search or MCE generation algorithm (cf. [43]) and replacing it with the new goal condition. However, unlike the traditional MCE algorithm, the fact that the model updates here are detected for an observation sequence gives rise to two interesting properties.

Proposition 3. *For a model pair $\mathbb{M}^\alpha$, if a model update pair $\mathcal{E}$ is VII for an observation sequence $O$, then it must be a VII for any observation sequence $\hat{O}$ that contains $O$ as a prefix. However, the reverse is not true.*

This proposition is rather straightforward given the fact that for a model update pair to be VII, it needs to eliminate all failing plans the human may consider. Any possible failing plan that the human may consider after committing to additional steps must be a subset of this set. However, a VII that merely removes a subset need not work for the original set.

Proposition 4. *For a model pair $\mathbb{M}^{\alpha}$, if a model update pair $\mathcal{E}$ is MII for an observation sequence $O$, and $\widetilde{\mathcal{E}}$ an MII for an observation sequence $\hat{O}$ that contains $O$ as a prefix, then we must have*

$$|\widetilde{\mathcal{E}}^{+}| + |\widetilde{\mathcal{E}}^{-}| \leq |\mathcal{E}^{+}| + |\mathcal{E}^{-}|$$

This proposition can be proved by following a similar line of reasoning as the previous proposition. This brings up an interesting question about trade-off, is it better to just generate a VII in advance that works for all possible actions the human can take in the beginning and just use it when the agent has enough confidence or is it better to wait until the point of failure and calculate an MII [1]. We will be evaluating this trade-off empirically in our evaluation. For the running example, the MII at every point involves informing the human about all the cells with the oil spill.

## 6 PRE-EMPTIVE INTERVENTION

Looking back at the running example (and Figure 2), the system is only confident about failure after step 4. At step 4, it can give information about the spilled oil and hopefully stop the human from making the mistake. As mentioned earlier, this would also leave the human stuck inside the room. They can no longer go back out of the room and switch on the other stove. If the goal of the system was to empower the human to actually achieve the goal, as opposed to just preventing them from making a mistake, the information should have been provided much earlier.

This could be the case with many non-ergodic domains, where waiting until the system is confident could result in the human receiving the information too late to actually be used effectively. Instead of merely reasoning about the likelihood of the human making mistake, it needs to reason about the expected costs involved and use it to drive the decision-making. A natural way to express this reasoning problem would be in terms of a POMDP.

Framed from the point of view of the agent, the actions of the POMDP are limited to a NOOP action (i.e., the agent doesn't intervene), and the action to provide the model updates ($a_{\mathcal{E}}$). The state here includes the current task state as evaluated by the agent, the current task state as evaluated by the human, the human's current belief about the task model, and finally, the plan that is currently being pursued by the user. In our setting, all elements except the current plan are considered observable. The transition function for NOOP action simply involves the execution of the human action and updating both the agent and human's estimate of the state. On the other hand, the choice to give model updates, would update the human model and maybe also the plan they might pursue. The likelihood of the plan selection can be determined by the noisy rational model described before. The cost function could correspond to a failure cost $c_f$ if the system transitions to a state that corresponds to a failure state, cost of model update $c_{\mathcal{E}}(s)$ if the system provides one at state $s$, and zero for all other transitions (with $c_f >> c_{\mathcal{E}}(s)$ for all $s$). Note that we will use a more expansive definition of a failure state than a state obtained by the application of an action whose preconditions are not met. In fact, we will consider any state from which the human goal cannot be achieved to be a failure state and also treat it as an absorbing state. We will denote this POMDP as $\mathcal{M}^{(H,\alpha)}$.

Even with the advances in POMDP solvers, exactly solving this problem may not be feasible or practically effective. In fact, even building this model is an expensive process given the need to identify all the states from which the goal is not reachable. However, as we will see, the setting lends itself to be approximated easily.

First, instead of considering individual plans as part of the state, we will aggregate them into a binary variable that represents whether the human is pursuing a plan currently that is bound to fail or not fail ($F$ or $\neg F$). We will use the notation $S_t$ to capture the observable part of the state at a given time step $t$. Finally, we will leverage the intuition that once the model updates are given, the human is guaranteed to succeed. Thus there is no reason to ever repeat this information.

At any timestep $t$ and state $S_t$, let the probability of the human following plan that fails be $\mathcal{P}_F$ (shortened from $\mathcal{P}_F(O_t, \mathbb{M}^{\alpha})$ for convenience). For the current step, the cost of giving a model update will be just $c_{\mathcal{E}}$. For NOOP, in cases where the human may be pursuing a plan that will fail, we will look at what the next potential states are and then try to approximate the cost from that state. If no failure has happened in the next state, we will approximate the future cost by using ($c_{\mathcal{E}}$). This is equivalent to saying that if no error occurs in the next step, the agent will take the safer option of giving a model update. The cost for NOOP action is given as

$$\hat{C}_{NOOP}(S_t) = \mathcal{P}_F \times \sum_{S_{t+1}} P(S_{t+1}|S_t) \times C(S_{t+1})$$

In the above equation, $P(S_{t+1}|S_t)$ effectively considers the probability of transition under each possible plan, along with the likelihood of the plan. However, the formulation provided in Section 4 provides us with the tools to calculate it without explicitly enumerating it over all plans. Note that for states $S_{t+1}$ which are failure states, the cost $C(S_{t+1})$ will automatically be $c_f$ and $c_{\mathcal{E}}(S_{t+1})$ for the rest of the states. We can ignore the $(1 - \mathcal{P}_F)$ term, since there is no cost associated with following a plan that will not fail.

We can show that the above cost calculation is a cautious approximation, as it always overapproximates the cost of performing a NOOP action. More formally, we can state this as

Theorem 1. *For any belief state $B^{(H,\alpha)}$ of the original POMDP, we have $\hat{C}_{NOOP}(S) \geq Q^{*}(B^{(H,\alpha)}, NOOP)$, where $S$ is the common observable part shared by all states with non-zero probability in $B^{(H,\alpha)}$, and $Q^{*}$ is optimal Q-value for the corresponding belief space MDP.*

Proof Sketch. This theorem can be proved easily by considering two facts. First, the aggregation maintains all the individual probabilities (as proved in the previous section). Secondly, we can consider a QMDP approximation [28] of the POMDP. For QMDP, we can already see that for NOOP, the Q-value cost will be higher. Secondly, we see that for the MDP estimates where the current state corresponds to a failing plan, the optimal plan would always involve waiting until the failure step and then providing the model updates. This will be smaller than the value provided here. □

[1] One could in theory also argue for a rather conservative approach of always giving a VII upfront. However, in cases where the actual likelihood of the human making a mistake is low, this will not only place an unnecessary cognitive load on the human but could also result in humans losing trust in the system and potentially ignoring future warnings.

Our approximate decision-making procedure would involve choosing the explanation action, whenever $\hat{C}_{NOOP}(S) > c_{\mathcal{E}}$. Given the results of Theorem 1 and the fact that $Q^*(B^{(H,\alpha)}, a_{\mathcal{E}}) = c_{\mathcal{E}}$, we can guarantee that there will never be a state where the optimal policy would choose to perform the model update action ($a_{\mathcal{E}}$) and the approximate method would choose to perform a NOOP action. This further means that our method is always guaranteed to provide model updates before or at the same time as the optimal policy.

## 7 RELATED WORK

Designing effective pro-active decision-support/assistive systems requires the creation of adaptive systems that can seamlessly comprehend the user's requirements, goals, and limitations. Then be able to directly use that information to come up with suggestions and even corrective actions to help the user achieve their intended objective. As such many of the early works in this direction have focused on the problem of tracking user actions and identifying goals, and providing assistance when necessary.

There currently exists a rich set of works on activity, plan, and goal recognition [51]. In particular, the methods discussed in the paper are particularly connected to the goal-recognition literature [36, 41] as defined within the context of classical planning. Some works have tried to incorporate the problem of goal recognition directly into the assistance framework (cf. [3, 8, 15]). There are also frameworks like CIRL [19], where there is an expectation that humans might take an active role in helping communicate their objectives. On the other hand works in active goal recognition [39], looks at endowing the observer/recognizer agency to improve recognition. There have also been similar systems designed to support people with cognitive disabilities [11]. There is also a wider literature on intervention in both adversarial and cooperative settings [25, 56]. Our proposed algorithm can also be seen as a special case of the intervention problem defined by Weerawardhana et al. [55].

However, one set of scenarios where knowledge asymmetry has been explored to a degree is within the problem of providing decision support during the planning phase. Here the human is in the middle of coming up with a plan of action and using a specific decision support interface to formalize their plan. The agent could keep track of this plan and then provide suggestions on other courses of action to take and even provide explanations. Some prominent examples of such systems are the RADAR decision-support systems [33, 38, 49]. Such systems have also been explored in the context of risk management [2, 40]. One important example from this group of work is the one used in enterprise settings, which makes use of diverse planning.

In the context of human-robot Interaction, related approaches have been investigated in the context of shared autonomy [37]. In these cases, the control of a system is shared between the user and some form of automated decision-making system. In many of these systems, inferring the user goal is important to ensure the system is helping the user effectively [24]. Other relevant works in this direction include human-robot joint actions [12, 13] and coordination [48]. Most of these works involve social interaction in which two or more individuals/agents coordinate their actions in a shared space to effect a change in the environment, the emphasis of our research is on preventing the human actor from performing actions that could lead to failure. A pivotal element of the proposed proactive assistance system is its ability to reason about human mental states. Consequently, our work aligns closely with the Theory of Mind concepts in AI planning [1, 14, 23]. This includes the concept of Perspective Taking—a human ability that enables individuals to see things from another's viewpoint [32].

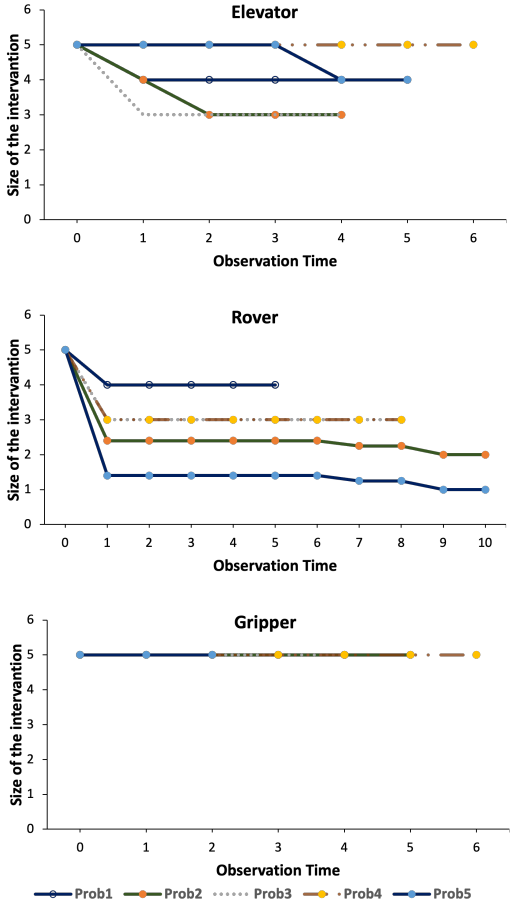


**Figure 3: A plot showing how the average size of the minimal intervention varies with the observation length.**

## 8 EVALUATION

For empirical evaluation, our primary goal was to evaluate the two proposed approaches over a set of standard IPC benchmarks. The first method (M1) merely assesses the likelihood of goal failure through observation and intervenes when the likelihood of failure is higher, while the second method (M2) adopts a preemptive approach to identifying points at which to intervene. Through this evaluation, we are interested in testing three main hypotheses.

**H1** M2 can help prevent failures that might happen under M1.

**H2** The use of satisficing planners for approximating goal likelihood will result in lower runtime without serious degradation in performance with regard to the failure detection step.

| Domains | O | F.S | Optimal | | | | | Satisficing | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | M1 | | | M2 | | M1 | | | M2 | |
| | | | Int. Step | FR | Time(s) | Int. Step | Time(s) | Int. Step | FR | Time(s) | Int. Step | Time(s) |
| Elevator | 5.40 | 4.04 | 3.2 ± 1.8 | 3.6/5 | 3.2 ± 2.5 | 0.95 ± 0.8 | 77.2 ± 62.3 | 3.7 ± 1.7 | 4.4/5 | 3.6 ± 2.3 | 0.95 ± 0.8 | 72.9 ± 61.0 |
| Rovers | 9.24 | 7.04 | 6.9 ± 1.6 | 4.8/5 | 16.6 ± 7.0 | 2.9 ± 0.8 | 1800.7 ± 2036.7 | 6.1 ± 1.9 | 3.2/5 | 15.3 ± 9.3 | 2.3 ± 1.1 | 674.7 ± 659.5 |
| Gripper | 5.60 | 3.44 | 3.2 ± 1.6 | 4.8/5 | 6.1 ± 7.1 | 1.2 ± 0.8 | 154.4 ± 178.7 | 3.0 ± 1.7 | 4/5 | 3.1 ± 2.8 | 0.8 ± 0.7 | 104.5 ± 82.7 |
| Zenotravel | 5.96 | 4.16 | 3.5 ± 1.6 | 4.2/5 | 7.1 ± 5.8 | 0.5 ± 0.4 | 1552.5 ± 2091.6 | 2.4 ± 1.7 | 2.4/5 | 3.9 ± 4.7 | 0.3 ± 0.5 | 903.8 ± 735.7 |

**Table 1: Comparison of performance metrics for the methods (M1 and M2) using both optimal and satisficing planners across different domains with varying problems. Here O corresponds to the average observation length; F.S - the average step at which the observation would have resulted in a failure; FR - the ratio of instances in which the method failed to prevent the human from taking a step that results in failure; Int. Step - the average step number in which the method intervened (along with the std deviation). Finally, time reports the time taken for the approach as a whole.**

**H3** We expect the model update size to go down with observation length.

The code for the evaluation and the supplementary files can be found at https://github.com/cglrtrgy/Help.

**Setting:** We evaluated our method on four IPC domains. For each domain, the original domain description acted as the agent model, while we created the human domain model by randomly deleting five preconditions and delete effects. We created five distinct human models for each domain. Next, we selected five problems per domain. Thus per domain, we had 25 unique pairs of human and agent models (please note that in our terminology, a model contains both the domain and problem information). The optimal planner consisted of the FastDownward implementation [22] of A* search with hmax heuristic and used Lama as our satisficing planner. The observations were also generated by a satisficing planner (lazy-greedy search). All experiments were performed on a machine powered by an Intel Processor running at 3.10 GHz and with 128 GB RAM [35]. In our experiment, the planners were allowed to operate without any time or memory constraints, and we only considered unit cost actions. The cost of failure was taken to be double that of giving a model update.

**Results:** Table 1 gives an overview of the computational characteristics of the proposed methods and also provides all the information relevant to the first two hypotheses. Due to space limitations, we have only reported the average across all 25 instances, but you can find the full data in the supplementary file. Here for each problem instance, we present the results averaged across all five pairs of human and agent models. Specifically, for H1, we are interested in determining whether M2 provides any advantages over M1. Here we see clearly that, the failure rate of M1 (no of times the method intervened after the human made an error) is pretty high throughout all the domains. We skip reporting the failure rate for M2 because it never failed to catch a potential failure. These results support H1. However, we do see that M2 does take more time than M1 (here, the time doesn't involve the time taken for model update generation). This increase in time could be explained by the large branching factors of IPC domains and the fact that you are calculating the probability of each potential outcome.

This brings us to H2. Here we do see that even when using a satisficing planner, M2 never allowed the user to take a step that fails. Additionally, using a satisficing planner does result in some reduction in the time taken. The domain with the most significant reduction in time was the Rover domain. This again supports hypothesis H2.

For H3, we considered three of the above domains and plotted how the minimal model update size changed with observation length. The graph plots the average length of minimal interventional information or MII across the five human agent model pairs for each instance. In the domains Rover and Elevator, we saw that there was a reduction in the minimal intervention information size, with the reduction being most significant in Rover. However, we do not see the same reduction in Gripper, where MII size stayed the same across all observation sizes and problem instances. This might be explained by the fact that Gripper is an extremely compact domain and removing any model component results in an invalid domain. While these results do give some support for H3, it does point to the need to run more evaluation to better characterize how the model update size changes with observation length. Finally, we say that the average time taken across all domains was 278.7 seconds with a standard deviation of 200.5. The full breakdown of the time taken is provided in the supplementary file.

## 9 CONCLUSION AND FUTURE WORK

The work presents the first-of-its-kind pro-active assistance system which is able to identify potential human errors in the presence of knowledge asymmetry. We show how we can turn the problem of finding the probability of failure into a goal recognition problem. We additionally build on this basic formalism to support pre-emptive assistance to prevent humans from taking action that may result in them never reaching their goal. We also looked at how we can use model-space search to identify what information should be provided to the user to prevent such errors. In the future, we would like to look at problems where the assistive agent may be embodied or have limitations on how they could intervene. This would mean the agent would need to identify an error early enough that they can actually intervene. In future research, we plan to delve deeper into analyzing the algorithmic complexity of this problem.

## ACKNOWLEDGMENTS

Sarath Sreedharan's research is supported in part by grant NSF 2303019.

# REFERENCES


[1] Chris L Baker and Joshua B Tenenbaum. 2014. Modeling human plan recognition using Bayesian theory of mind. *Plan, activity, and intent recognition: Theory and practice* 7 (2014), 177–204.

[2] George Baryannis, Samir Dani, Sahar Validi, and Grigoris Antoniou. 2019. Decision support systems and artificial intelligence in supply chain risk management. *Revisiting supply chain risk* (2019), 53–71.

[3] Gregor Behnke, Benedikt Leichtmann, Pascal Bercher, Daniel Holler, Verena Nitsch, Martin Baumann, and Susanne Biundo. 2017. Help me make a dinner! Challenges when assisting humans in action planning. In *2017 international conference on companion technology (ICCT)*. IEEE, 1–6.

[4] Christine Bigby, Jacinta Douglas, and Lisa Hamilton. 2023. Overview of literature about enabling risk for people with cognitive disabilities in context of disability support services. (2023).

[5] Michael Bruss and Alexander Pfalzgraf. 2016. Proactive Assistance Functions for HMIs through Artificial Intelligence. *ATZ worldwide* 118, 12 (2016), 40–43.

[6] Anthony R Cassandra. 1998. A survey of POMDP applications. In *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, Vol. 1724.

[7] Tathagata Chakraborti, Gordon Briggs, Kartik Talamadupula, Yu Zhang, Matthias Scheutz, David Smith, and Subbarao Kambhampati. 2015. Planning for serendipity. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5300–5306.

[8] Tathagata Chakraborti, Gordon Briggs, Kartik Talamadupula, Yu Zhang, Matthias Scheutz, David Smith, and Subbarao Kambhampati. 2015. Planning for serendipity. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5300–5306.

[9] Tathagata Chakraborti, Sarath Sreedharan, Sachin Grover, and Subbarao Kambhampati. 2019. Plan explanations as model reconciliation–an empirical study. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Ieee, 258–266.

[10] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. *arXiv preprint arXiv:1701.08317* (2017).

[11] Yi Chu, Young Chol Song, Richard Levinson, and Henry Kautz. 2012. Interactive activity recognition and prompting to assist people with cognitive disabilities. *Journal of Ambient Intelligence and Smart Environments* 4, 5 (2012), 443–459.

[12] Aurelie Clodic and Rachid Alami. 2021. What Is It to Implement a Human-Robot Joint Action? *Robotics, AI, and humanity: Science, ethics, and policy* (2021), 229–238.

[13] Aurélie Clodic, Rachid Alami, and Raja Chatila. 2014. Key elements for joint human-robot action. In *Robo-Philosophy*, Vol. 273. IOS Press Ebooks, 23–33.

[14] Fabio Cuzzolin, Alice Morelli, Bogdan Cirstea, and Barbara J Sahakian. 2020. Knowing me, knowing you: theory of mind in AI. *Psychological medicine* 50, 7 (2020), 1057–1061.

[15] Alan Fern, Sriraam Natarajan, Kshitij Judah, and Prasad Tadepalli. 2014. A decision-theoretic model of assistance. *Journal of Artificial Intelligence Research* 50 (2014), 71–104.

[16] Richard E Fikes and Nils J Nilsson. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence* 2, 3-4 (1971), 189–208.

[17] Karl Friston, Philipp Schwartenbeck, Thomas FitzGerald, Michael Moutoussis, Timothy Behrens, and Raymond J Dolan. 2014. The anatomy of choice: dopamine and decision-making. *Philosophical Transactions of the Royal Society B: Biological Sciences* 369, 1655 (2014), 20130481.

[18] Camille Grange, Izak Benbasat, and Andrew Burton-Jones. 2019. With a little help from my friends: Cultivating serendipity in online shopping environments. *Information & Management* 56, 2 (2019), 225–235.

[19] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. 2016. Cooperative inverse reinforcement learning. *Advances in neural information processing systems* 29 (2016).

[20] Helen Harman and Pieter Simoens. 2020. Action graphs for proactive robot assistance in smart environments. *Journal of Ambient Intelligence and Smart Environments* 12, 2 (2020), 79–99.

[21] Tianzhi He, Farrokh Jazizadeh, and Laura Arpan. 2022. AI-powered virtual assistants nudging occupants for energy saving: proactive smart speakers for HVAC control. *Building Research & Information* 50, 4 (2022), 394–409.

[22] Malte Helmert. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26 (2006), 191–246.

[23] Mark K Ho, Rebecca Saxe, and Fiery Cushman. 2022. Planning with theory of mind. *Trends in Cognitive Sciences* 26, 11 (2022), 959–971.

[24] Shervin Javdani, Siddhartha S Srinivasa, and J Andrew Bagnell. 2015. Shared autonomy via hindsight optimization. *Robotics science and systems: online proceedings* 2015 (2015).

[25] Gal A Kaminka, David V Pynadath, and Milind Tambe. 2001. Monitoring deployed agent teams. In *Proceedings of the fifth international conference on Autonomous agents*. 308–315.

[26] Steven M LaValle. 2006. *Planning algorithms*. Cambridge university press.

[27] Minha Lee, Sander Ackermans, Nena Van As, Hanwen Chang, Enzo Lucas, and Wijnand IJsselsteijn. 2019. Caring for Vincent: a chatbot for self-compassion. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[28] Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. 1995. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*. Elsevier, 362–370.

[29] Filippo Menczer, W Nick Street, Narayan Vishwakarma, Alvaro E Monge, and Markus Jakobsson. 2002. IntelliShopper: A proactive, personal, private shopping assistant. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*. 1001–1008.

[30] Merehau C Mervin, Wendy Moyle, Cindy Jones, Jenny Murfield, Brian Draper, Elizabeth Beattie, David HK Shum, Siobhan O'Dwyer, and Lukman Thalib. 2018. The cost-effectiveness of using PARO, a therapeutic robotic seal, to reduce agitation and medication use in dementia: findings from a cluster–randomized controlled trial. *Journal of the American Medical Directors Association* 19, 7 (2018), 619–622.

[31] Christian Meurisch, Maria-Dorina Ionescu, Benedikt Schmidt, and Max Mühlhäuser. 2017. Reference model of next-generation digital personal assistant: integrating proactive behavior. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. 149–152.

[32] Grégoire Milliez, Matthieu Warnier, Aurélie Clodic, and Rachid Alami. 2014. A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management. In *The 23rd IEEE international symposium on robot and human interactive communication*. IEEE, 1103–1109.

[33] Aditya Prasad Mishra, Sailik Sengupta, Sarath Sreedharan, Tathagata Chakraborti, and Subbarao Kambhampati. 2019. Cap: A decision support system for crew scheduling using automated planning. *Naturalistic Decision Making* (2019).

[34] George E Monahan. 1982. State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms. *Management science* 28, 1 (1982), 1–16.

[35] Intel Xeon Processor. 2016. E5-2630 v4.

[36] Miguel Ramírez and Hector Geffner. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 24. 1121–1126.

[37] Mario Selvaggio, Marco Cognetti, Stefanos Nikolaidis, Serena Ivaldi, and Bruno Siciliano. 2021. Autonomy in physical human-robot interaction: A brief survey. *IEEE Robotics and Automation Letters* 6, 4 (2021), 7989–7996.

[38] Sailik Sengupta, Tathagata Chakraborti, Sarath Sreedharan, Satya Gautam Vadlamudi, and Subbarao Kambhampati. 2017. Radar—a proactive decision support system for human-in-the-loop planning. In *2017 AAAI Fall Symposium Series*.

[39] Maayan Shvo and Sheila A McIlraith. 2020. Active goal recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9957–9966.

[40] Shirin Sohrabi, Anton Riabov, Michael Katz, and Octavian Udrea. 2018. An AI planning solution to scenario generation for enterprise risk management. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[41] Shirin Sohrabi, Anton V Riabov, and Octavian Udrea. 2016. Plan Recognition as Planning Revisited.. In *IJCAI*. New York, NY, 3258–3264.

[42] Sarath Sreedharan, Pascal Bercher, and Subbarao Kambhampati. 2022. On the computational complexity of model reconciliations. In *31st International Joint Conference on Artificial Intelligence, IJCAI 2022*. International Joint Conferences on Artificial Intelligence, 4657–4664.

[43] Sarath Sreedharan, Tathagata Chakraborti, and Subbarao Kambhampati. 2021. Foundations of explanations as model reconciliation. *Artificial Intelligence* 301 (2021), 103558.

[44] Sarath Sreedharan, Subbarao Kambhampati, et al. 2018. Handling model uncertainty and multiplicity in explanations via model reconciliation. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 28. 518–526.

[45] Sarath Sreedharan, Anagha Kulkarni, David E Smith, and Subbarao Kambhampati. 2021. A Unifying Bayesian Formulation of Measures of Interpretability in Human-AI Interaction.. In *IJCAI*. 4602–4610.

[46] Sarath Sreedharan, Siddharth Srivastava, David Smith, and Subbarao Kambhampati. 2019. Why can't you do that hal? explaining unsolvability of planning tasks. In *International Joint Conference on Artificial Intelligence*.

[47] Yu Sun, Nicholas Jing Yuan, Yingzi Wang, Xing Xie, Kieran McDonald, and Rui Zhang. 2016. Contextual intent tracking for personal assistants. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 273–282.

[48] Kartik Talamadupula, Gordon Briggs, Tathagata Chakraborti, Matthias Scheutz, and Subbarao Kambhampati. 2014. Coordination in human-robot teams using mental modeling and plan recognition. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2957–2962.

[49] Karthik Valmeekam, Sarath Sreedharan, Sailik Sengupta, and Subbarao Kambhampati. 2022. RADAR-X: An Interactive Mixed Initiative Planning Interface Pairing Contrastive Explanations and Revised Plan Suggestions. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 32.

508–517.

[50] Cora Van Leeuwen, Annelien Smets, An Jacobs, and Pieter Ballon. 2021. Blind spots in AI: the role of serendipity and equity in algorithm-based decision-making. *ACM SIGKDD Explorations Newsletter* 23, 1 (2021), 42–49.

[51] Mor Vered, Reuth Mirsky, Ramon Fraga Pereira, and Felipe Meneguzzi. 2022. Advances in Goal, Plan and Activity Recognition. *Frontiers in Artificial Intelligence* 5 (2022), 861669.

[52] Pooja Viswanathan, James J Little, Alan K Mackworth, and Alex Mihailidis. 2012. An intelligent powered wheelchair for users with dementia: case studies with NOAH (navigation and obstacle avoidance help). In *2012 AAAI Fall Symposium Series*.

[53] Francesco Vona, Emanuele Torelli, Eleonora Beccaluva, and Franca Garzotto. 2020. Exploring the potential of speech-based virtual assistants in mixed reality applications for people with cognitive disabilities. In *Proceedings of the International Conference on Advanced Visual Interfaces*. 1–9.

[54] Nagagopiraju Vullam, Sai Srinivas Vellela, Venkateswara Reddy, M Venkateswara Rao, Khader Basha SK, and D Roja. 2023. Multi-Agent Personalized Recommendation System in E-Commerce based on User. In *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. IEEE, 1194–1199.

[55] Sachini Weerawardhana, Darrell Whitley, and Mark Roberts. 2022. Models of Intervention: Helping Agents and Human Users Avoid Undesirable Outcomes. *Frontiers in Artificial Intelligence* 4 (2022), 723936.

[56] David E Wilkins, Thomas J Lee, and Pauline Berry. 2003. Interactive execution monitoring of agent teams. *Journal of Artificial Intelligence Research* 18 (2003), 217–261.

[57] Neil Yorke-Smith, Shahin Saadati, Karen L Myers, and David N Morley. 2012. The design of a proactive personal agent for task management. *International Journal on Artificial Intelligence Tools* 21, 01 (2012), 1250004.

[58] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. 2017. Plan explicability and predictability for robot task planning. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 1313–1320.