Who Am I Dealing With? Explaining the Designer's Hidden Intentions

Turgay Caglar
Colorado State University
Fort Collins, USA
turgay.caglar@colostate.edu

Sarath Sreedharan
Colorado State University
Fort Collins, USA
sarath.sreedharan@colostate.edu

Mor Vered
Monash University
Melbourne, Australia
mor.vered@monash.edu

ABSTRACT

Explainable AI (XAI) methods are generally seen as tools that allow users a greater level of visibility into why certain decisions were made by an AI system or agent. However, by the very choice of current works to focus on merely explaining why the AI system chose to perform an action in their environment, the explanation is withholding any information about the role played by the *designer* of said system and environment in determining the final behavior. This information could be particularly significant when the underlying designer objectives may differ from those of the user. In this paper we propose a new explanation generation paradigm, built on the concept of model reconciliation, and show how it can support the generation of explanations that include the designer's goals. We define and study the formal properties of this new form of explanation and introduce an algorithm to generate it over a classical planning domain. We evaluate how this new explanation influences user performance, understanding and trust in an AI agent and further instantiate the new algorithm on standard planning benchmarks to evaluate its computational characteristics.

KEYWORDS

XAI, Transparency, Designer Explanations, Model Reconciliation

ACM Reference Format:
Turgay Caglar, Sarath Sreedharan, and Mor Vered. 2025. Who Am I Dealing With? Explaining the Designer's Hidden Intentions. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Human agent collaboration enables a varied, distributed set of actors to work together to address problems of greater complexity than those able to be addressed by each actor alone. However, the field of user-agent interaction presents several challenges, including issues related to trust, utilization, and varying degrees of reliance, which can hinder effective collaborations [8, 24, 25]. The problem of eXplanaining AI (XAI), often equated with interpretable AI [11], has been developed to address these challenges by enhancing users' understanding of AI systems and fostering more optimal interactions [2, 19]. XAI approaches have aimed to improve people's understanding of the agent model, help people recognize model uncertainty, and support people's calibrated trust in the agent [15, 27].

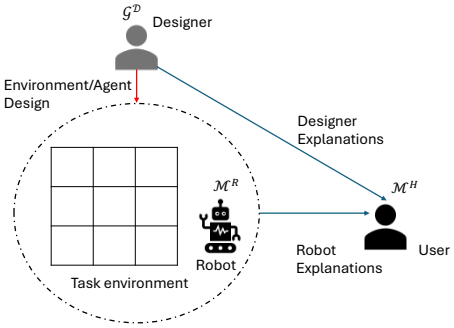

Figure 1: An overview of the interactions captured in our framework. The designer tries to modify the environment so that the robot's behavior achieves its underlying goal $\mathcal{G}^{\mathcal{D}}$.

One XAI approach that aims to bridge this gap is through applying model-reconciliation, ascribing the agents with an approximation of the human's task and goal models [21]. In this approach, the agent explains its actions by leveraging the differences between its own model and the human's mental model of the agent. By reconciling the differences between the agent's model and the human's perception of that model, this method aims to align the human's expectation with the outcomes of the agent's behavior.

However, we posit that focusing on explaining the *agent model*, be it through model reconciliation or otherwise, should not be the goal and may even (unintentionally) create user deception. There is a third actor about whose intentions the user needs to reason. That is the *designer*, the actor who created the AI agent and made key choices about it's design and performance (Figure 1). It is rare for AI agents to be deployed and operate in a completely uncontrolled environment. In most cases, at least some aspects of the environment would have been controlled or designed to promote certain forms of agent behavior. Obviously, such a design process would also be applied to the agent itself. When an end-user comes in contact with an AI agent and asks it to achieve certain objectives, the responding agent actions would be heavily dependent on these prior design choices.

For the most part the user is oblivious to the design decisions taking place behind the scenes. And while the designer's intentions may overlap with the agent's professed intentions (for example a search and rescue robot's intention is the same as the designers' intention for it - finding the human), they also may not. Consider recommendation systems: the agent's professed intention is to recommend a product that is similar to other products that the user likes and would therefore enjoy. On the other hand the designer's intention is for the user to ultimately spend more money, thereby

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

increasing its utility. It is these goals that are often consciously or unconsciously hidden from the user through the mask of the agent. Same as in deception by misdirection, in which a person is deceived by focusing their attention in the wrong place - when explaining the robot's actions rather than the designer's, these explanations mask the true intention by shifting the user attention to reason about the acting agent [18]. In effect, the explanations people are receiving are not the explanations they need.

To bridge this gap we propose a new explanation framework which explains both designer intentions and acting agent actions. We begin by formalising a multi-actor explanation framework, this time considering the additional aspect of the designer/stakeholder (Section 4). We then instantiate our framework on the classical planning Sokoban environment and empirically evaluate the explanation generation performance and efficiency (Section 5). Finally, we conduct a proof-of-concept user study in which users are presented with designer explanations (Section 6). We discuss the results and propose ideas for future work.

2 BACKGROUND AND RELATED WORK

As AI applications become more widespread and even deployed in safety-critical settings, there is increasing recognition that these systems need to be capable of explaining their decisions [12]. While some of the early works in explanations go back to expert systems [6, 22], the more recent interest has been particularly spurred by the inscrutability of the state-of-the-art AI models [15]. Even in a subfield of AI, like planning, we see a pretty large number of works related to explanations (cf. [1, 4]). However, recent works have also highlighted how explanations could lead to misuse of the system. This includes how even simple explanations could cause people to place unwarranted trust in the system [23] and even accept system decisions against their best interest [16]. Despite the rapid advancements in the field, most explanation efforts remain focused on a straightforward dyadic settings [21]. To the best of our knowledge, no other works in XAI has looked at incorporating the influence the designer has on the final behavior.

2.1 Background

We consider a setting where the decision-making problem can be best represented as a classical planning problem. While the problem of designer goals and explanations pertains to all representations, we found the classical planning domain as the most adapt at providing a clear, relatable proof-of-concept. Equivalently we rely on the classical STRIPS [9] planning model of $\mathcal{M} = \langle F, A, I, G \rangle$, where F represents the set of propositional fluents, A the set of actions, $I \subseteq F$ the initial state, and G the goal specification.The state space corresponding to the planning problem is defined using the fluent set, such that, each state s can be uniquely identified by the set of fluents that are true in that state, $s \subseteq F$. Each action $a \in A$, is further defined by a set of preconditions ($pre(a) \subseteq F$), add effects ($add(a) \subseteq F$), delete effects ($del(a) \subseteq F$), and a cost c_a. The result of executing action a in state s is given by the transition function $\delta^{\mathcal{M}} : S \times A \rightarrow 2^F$, such that:

$$\delta^{\mathcal{M}}(s, a) = \begin{cases} (s \setminus del(a)) \cup add(a) \text{ if } pre(a) \subseteq s \\ \textit{Undefined} \text{ Otherwise} \end{cases}$$

A solution to a planning problem is a plan, a sequence of actions, whose execution satisfies G. More formally, an action sequence $\pi = \langle a_1, ..., a_k \rangle$ is said to be a plan, if $\delta^{\mathcal{M}}(I, \pi) \models G$. The cost of a plan $\pi = \langle a_1, ..., a_k \rangle$ for a model $\mathcal{M}$, is given as the accumaltive sum of the cost of all actions in that plan, $C^{\mathcal{M}}(\pi) = c_{a_1} + ... + c_{a_k}$. And a plan is said to be optimal if no other plan exists at a lower cost. We will represent the cost of the optimal plan for a model $\mathcal{M}$ as $C^*_{\mathcal{M}}$.

Since our solution approaches leverage model space search, we will further represent models using a set of propositional model parameters. The space of all parameters needed to capture models that use fluents F, action labels A and cost set $\mathbb{C}$ is given as

$$\mathcal{F} = \{\textit{init-has-f} \mid f \in F\} \cup \{\textit{goal-has-f} \mid f \in F\} \cup \bigcup_{a \in A} \{\textit{a-has-cost-c} \mid c \in \mathbb{C}\} \cup \{\textit{a-has-prec-f}, \textit{a-has-add-f}, \textit{a-has-del-f} \mid f \in F\}.$$

We will use the parameterization function Γ to convert a given model to its parameterized form, mapping each model component to its corresponding proposition in $\mathcal{F}$. We will use the function Γ^{-1} to obtain the model corresponding to a given set of model parameters. In this context, a model update takes the form $\mathcal{E} = \langle \mathcal{E}^+, \mathcal{E}^- \rangle$, where $\mathcal{E}^+ \subseteq \mathcal{F}$ are the set of model parameters to be turned true and $\mathcal{E}^- \subseteq \mathcal{F}$ are the ones to be turned false. Now the model obtained by applying this model update on $\mathcal{M}$, is given as

$$\mathcal{M} + \mathcal{E} = \Gamma^{-1}((\Gamma(\mathcal{M}) \setminus \mathcal{E}^-) \cup \mathcal{E}^+)$$

Where '+' is the model update operator.

3 RUNNING EXAMPLE

To illustrate our approach consider Figure 2. In this scenario, there are three actors; (1) *the operator* who is tasked with helping robots navigate from a start position to a goal location by choosing one of several possible routes; (2) the robot whose goal is to reach the flag safely and with the least possible actions. The robot can move one square at a time, but only in the direction its tires are facing. It can rotate by changing its facing direction 90 degrees at a time and each action has a uniform cost; and (3) the designer of the robot and environment. The designer's goal is for the robot (and therefore, operator) to view the ads. To achieve this, the designer positioned the robot facing left and placed the boxes to make the passage too narrow, rendering the purple path invalid. Note that the operator does not know that the robot is too wide to safely pass through narrow passages.

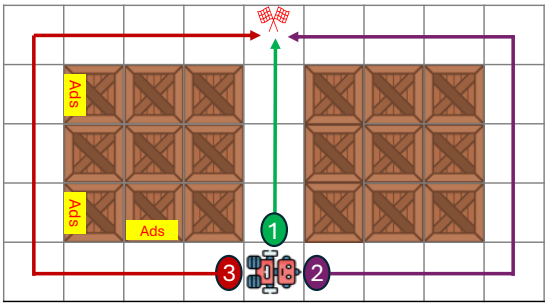

Figure 2: Example of decision making task

4 ENVIRONMENT DESIGN PROBLEM AND EXPLANATIONS

To solve the problem of designer explanations we adopt a two-level explanation strategy. First we explain why, given the current agent model and environment state, the current plan is the right course of action. Then, we explain how aspects of the current model which were under the designer's control, gave rise to agent behavior that helped achieve the designer's goal.

We begin by formally defining the underlying design problem that influenced the environment and gave rise to the particular agent behavior. We define the underlying designer goal as ($\mathcal{G}^{\mathcal{D}}$), the goal the designer wants the AI agent (henceforth referred to as the robot[1]) to achieve in the course of its operation (defined by a set of goals $\mathbb{G}^{\mathcal{R}}$). The designer has access to a set of actions ($\mathcal{A}^{\mathcal{D}}$) that they can employ to change both the environment and the robot. In our running example, $\mathbb{G}^{\mathcal{R}}$ ensures that the robot pass near 'Ads,' while $\mathcal{A}^{\mathcal{D}}$ includes placing boxes action (changing the environment) and rotating the robot's facing direction action (changing the robot).

The solution to the design problem is a sequence of designer actions that results in an environment where the behavior selected by the robot will satisfy the designer's goal. For convenience, we assume that the designer's actions only change the initial state of the robot and that the robot is an autonomous agent, using an independent optimal decision-making process to identify the optimal course of action to achieve its goal, given the environment. Note that the influence that the designer asserts on the agent is an indirect one, where they set up the environment so that the autonomous agent ends up also achieving the designer's hidden intent. This is a special case of mechanism design, which is an important topic within game theory.

We define the design problem, and solution, as follows:

Definition 1. *An environment design problem is characterized by the tuple, $\mathcal{DP} = \langle F, A^R, I^0, \mathcal{G}^{\mathcal{D}}, \mathcal{A}^{\mathcal{D}}, \mathbb{G}^{\mathcal{R}}, \Lambda \rangle$, where:*

- *$F, A^R, \mathbb{G}^{\mathcal{R}}$ - Fluents used in the robot model, robot actions, and potential goals the robot might come across, respectively.*
- *I^0 - Initial, unedited, state.*
- *$\mathcal{G}^{\mathcal{D}}, \mathcal{A}^{\mathcal{D}}$ - Designer goal and actions.*
- *Λ - Transition function related to the application of designer action in the initial state such that, $\Lambda : 2^F \times \mathcal{A}^{\mathcal{D}} \rightarrow 2^F$.*

Definition 2. *A solution to the environment design problem is a designer plan, which consists of a sequence of designer actions $\pi^{\mathcal{D}} = \langle a_1^{\mathcal{D}}, ..., a_k^{\mathcal{D}} \rangle$, such that the resultant initial state is $I^{\mathcal{D}} = \Lambda(I^0, \pi^{\mathcal{D}})$, and allows that for every $G' \in \mathbb{G}^{\mathcal{R}}$, the resultant robot model $\mathcal{M}' = \langle F, A^R, I^{\mathcal{D}}, G' \rangle$ is of the form that every plan π' optimal in $\mathcal{M}'$ satisfies $\mathcal{G}^{\mathcal{D}}$, i.e, $\delta^{\mathcal{M}'}(I^{\mathcal{D}}, \pi') \models \mathcal{G}^{\mathcal{D}}$.*

Following this definition, we can define the form of explanations we would want for this setting. We refer to these explanations as the *robot-designer explanations*. We use model reconciliation [3, 5, 20] as our base explanatory framework and extend it to support explaining the role of design choices. One aspect to remember here is that, as discussed before, the designer may want to hide the design influence from the user. Therefore an objective explainer can rarely be the designer or a system sanctioned or built by the designer. A more plausible role these systems could take would be that of an external, post-hoc system being employed by the user to make sense of the decision of an automated system. This necessarily restricts the information the explanation generation system might have access to. We generally adhere to information that can either be learned (possibly through observation of the robot and environment) or can be hypothesized directly from observed behavior.

The robot-designer explanations are based on the robot model ($\mathcal{M}^R$), the user's mental model of the robot ($\mathcal{M}^H$), the plan to be explained (π^R), the set of fluents whose values the designer could potentially influence ($\mathcal{F}^{\mathcal{D}} \subseteq F$), and the potential designer goal ($\mathcal{G}^{\mathcal{D}}$). As this is the first work considering designer intentions, we will focus on cases where a single fluent set and an individual hypothesis for the designer's goal are provided initially.

As discussed, our final explanation consists of two parts. (1) Explaining the robot behavior given the current environment, leveraging model reconciliation explanations [20]. Here, the explanation consists of information about the robot model, $\mathcal{M}^R$, which, when incorporated into the human model, $\mathcal{M}^H$, will allow the user to correctly evaluate the robot plan against the robot goal. (2) Model updates and counterfactual explanation [26]. The model updates will establish the fact that in the current model, the optimal plan always includes achieving the designer's goal, $\mathcal{G}^{\mathcal{D}}$. The counterfactual explanation will point out a set of initial state fluent values whose value change could result in an optimal plan for the robot goal that no longer achieves $\mathcal{G}^{\mathcal{D}}$. More formally, we define the explanation problem as follows:

Definition 3. *A robot-designer explanation problem is represented as a tuple $\mathcal{DEP} = \langle \mathcal{M}^R, \mathcal{M}^H, \pi^R, \mathcal{F}^{\mathcal{D}}, \mathcal{G}^{\mathcal{D}} \rangle$. Here, the primary components for the robot explanation include the robot model $\mathcal{M}^R$, the human model $\mathcal{M}^H$, and the robot plan π^R. The designer information being used includes the fluents that can be changed by the designer ($\mathcal{F}^{\mathcal{D}}$) and the designer goal ($\mathcal{G}^{\mathcal{D}}$). The $\mathcal{M}^R$ used here is assumed to be a result of an environment design process.*

This defines a robot-designer explanation problem, and now we can formally define what a solution to this problem, i.e., an explanation, looks like.

Definition 4. *For a given robot-designer explanation problem $\mathcal{DEP} = \langle \mathcal{M}^R, \mathcal{M}^H, \pi^R, \mathcal{F}^{\mathcal{D}}, \mathcal{G}^{\mathcal{D}} \rangle$, a valid explanation consists of tuple of the form $\mathbf{E} = \langle \mathcal{E}_R, E_{\mathcal{D}} \rangle$, where $\mathcal{E}_R$ is the robot explanation and $E_{\mathcal{D}}$ is the designer explanation of the form $E_{\mathcal{D}} = \langle \mathcal{E}_\mu, \mathcal{E}_\kappa \rangle$, such that the following conditions are met*

C1- *For $\mathcal{E}_R = \langle \mathcal{E}_R^+, \mathcal{E}_R^- \rangle$, $\mathcal{E}_\mu = \langle \mathcal{E}_\mu^+, \mathcal{E}_\mu^- \rangle$ and $\mathcal{E}_\kappa = \langle \mathcal{E}_\kappa^+, \mathcal{E}_\kappa^- \rangle$, are such that,*
  (1) *$\mathcal{E}_R^+ \subseteq \Gamma(\mathcal{M}^R) \setminus \Gamma(\mathcal{M}^H)$ and $\mathcal{E}_R^- \subseteq \Gamma(\mathcal{M}^H)$*
  (2) *$\mathcal{E}_\mu^+ \subseteq \Gamma(\mathcal{M}^R) \setminus \Gamma(\mathcal{M}^H)$ and $\mathcal{E}_\mu^- \subseteq \Gamma(\mathcal{M}^H)$*
  (3) *$\mathcal{E}_\kappa^+ \cup \mathcal{E}_\kappa^- \subseteq \mathcal{F}^{\mathcal{D}}$.*

C2- *The plan π^R is optimal for model $\mathcal{M}^H + \mathcal{E}_R$.*

C3- *For the model $\mathcal{M}^H + \mathcal{E}_R + \mathcal{E}_\mu$ there exists no optimal plan such that it doesn't satisfy $\mathcal{G}^{\mathcal{D}}$ and π^R is still optimal.*

C4- *for model $\mathcal{M}' = \mathcal{M}^H + \mathcal{E}_R + \mathcal{E}_\mu + \mathcal{E}_\kappa$, there exists an optimal plan π', such that $\delta^{\mathcal{M}'}(I', \pi') \not\models \mathcal{G}^{\mathcal{D}}$ and $C^{\mathcal{M}'}(\pi') \leq C^{\mathcal{M}'}(\pi')$.*

[1]Even though we refer to the AI agent as a robot, no part of this approach is limited to physically embodied agents.

In the case of the designer's explanation $\mathcal{E}_\mu$ captures the model update part, and $\mathcal{E}_\kappa$ the counterfactual part. Condition C1 sets the requirements for the model updates provided as part of the robot explanation to be consistent with $\mathcal{M}^R$. Similarly, it states that the designer explanation component should only consider changing fluents that are under the designer's control. The next three conditions, C2-C4, ensure that each explanation component meets its required purpose. The robot's explanation on its own shows why the current plan is optimal in the given environment. Returning to Figure 2, the robot's explanation would be: 'The robot is wide. The robot can only move through spaces that are wide.' This would help the user understand why the red path is optimal rather than the green path.

The first part of the designer's explanation will establish the fact that achieving the designer's goal will always be part of any optimal strategy in the given environment. The second part of the designer's explanation identifies some initial state fluents that the designer could influence. If changed, these fluents' values could allow for optimal plans that might not have met the designer's goals. This communicates the counterfactual cases where the designer's goals could have been avoided. Consider Figure 2, if the designer only changed the robot's facing direction from right to left without placing boxes on the left side of the green path, the robot would follow the green path which would be optimal.

Note that not all valid explanations may be equally effective or preferred by the user. After all, selectivity has been widely recognized as one of the central characteristics of preferred explanations [15]. As such, we need to minimize the amount of information passed to the user. Rather than optimizing the two components separately, we aim to minimize the total amount of information passed to the user.

Definition 5. *A given explanation pair, $\mathbf{E} = \langle \mathcal{E}_R, E_\mathcal{D} \rangle$, is considered a minimal explanation for the robot-designer explanation problem $\mathcal{DEP}$, if (a) it is valid explanation for $\mathcal{DEP}$, i.e., it meets the conditions listed in Definition 4 with respect to $\mathcal{DEP}$ and (b) there exists no other valid explanation $\hat{E} = \langle \hat{\mathcal{E}}_R, \hat{E}_\mathcal{D} \rangle$, such that*

$$|\hat{\mathcal{E}}_R^+| + |\hat{\mathcal{E}}_R^-| + |\hat{\mathcal{E}}_\mu^+| + |\hat{\mathcal{E}}_\mu^-| + |\hat{\mathcal{E}}_\kappa^+| + |\hat{\mathcal{E}}_\kappa^-| < |\mathcal{E}_R^+| + |\mathcal{E}_R^-| + |\mathcal{E}_\mu^+| + |\mathcal{E}_\mu^-| + |\mathcal{E}_\kappa^+| + |\mathcal{E}_\kappa^-|$$

We measure the cost associated with each explanation by the number of model updates it communicates. This measure was motivated both by its intuitiveness and generality. However, it is worth noting that we can easily associate an arbitrary cost function with each model update with minimal changes to the formulation and the explanation generation algorithm we will discuss next.

Before we discuss the algorithm we will first introduce some important properties of the robot-designer explanation.

Proposition 1. *For a given given robot-designer explanation problem $\mathcal{DEP} = \langle \mathcal{M}^R, \mathcal{M}^H, \pi^R, \mathcal{F}^\mathcal{D}, \mathcal{G}^\mathcal{D} \rangle$:*

1. *We can guarantee that a model update $\mathcal{E}_R$ exists, such that, both conditions C1 and C2 are met.*
2. *We can guarantee that a set of model update $\mathcal{E}_\mu$ exists, such that, both conditions C1 and C3 are met.*
3. *There might not exist a model update $\mathcal{E}_\kappa$ that meets C1, & C4.*

The first property arises from the fact that one can guarantee to meet $C1$ and $C2$, by just communicating the complete model $\mathcal{M}^R$. After all, $\mathcal{M}^R$ identified π^R as the optimal plan.

Similarly, since the current problem is the result of a design process, communicating the entire model will ensure that all possible optimal plans will satisfy the designer's goal, thus guaranteeing the existence of $\mathcal{E}_\mu$.

However, the counterfactual part of the designer explanation isn't guaranteed because the designer's goal could have been of the form that the initial state already guarantees its achievement (i.e., the solution to the original design problem was an empty sequence). Additionally, $\mathcal{F}^\mathcal{D}$ could be empty or might not have influenced the behavior of the plan.

The next property will deal with comparing robot explanations found as part of the minimal robot-designer explanation and minimally complete explanation (MCE) for the plan [20]:

Proposition 2. *Let $\mathcal{E}_R$ be part of a minimal robot-domain explanation for a problem $\mathcal{DEP} = \langle \mathcal{M}^R, \mathcal{M}^H, \pi^R, \mathcal{F}^\mathcal{D}, \mathcal{G}^\mathcal{D} \rangle$. We can guarantee that $|\mathcal{E}_R| \geq |\mathcal{E}_{MCE}|$, where $\mathcal{E}_{MCE}$ is the MCE for π^R given the models $\mathcal{M}^R$ and $\mathcal{M}^H$.*

This property follows from the fact that any robot explanation for $\mathcal{DEP}$ meets the criteria for an MCE and, as such, can be a candidate for an MCE explanation. Note that the robot explanation here is found as part of the overall minimal solution to $\mathcal{DEP}$. Since these are not required for MCE, it is possible to find smaller model updates set that meets the requirement for MCE but not those for the minimal explanation for $\mathcal{DEP}$.

Finally, another explanation property in the model reconciliation framework is that of *monotonicity* [20]. Namely, an explanation is non-monotonic if additional model updates can invalidate it. In our case, it means adding new model updates to the robot explanation and/or designer explanation component, causing it to violate conditions C2 and or C3.

Proposition 3. *A minimal robot-designer explanation E^* for a problem $\mathcal{DEP}$ need not be monotonic.*

The non-monotonicity of the robot explanation component directly follows the argument proposed through the concept of model reconciliation. New information about new preconditions (while missing information about other actions add effects), might cause one to think a plan previously thought to be optimal is now invalid (thus violating C2). Similar arguments can also be made for the model update part of the designer explanation. For the counterfactual component, C4 might have been satisfied by initial state changes because it now made new plans possible. However, additional changes to the initial state could invalidate those plans, hence violating C4.

4.1 Identifying Minimal Robot-Designer Explanation Set

To identify the minimal robot-designer explanation we will use model space search which has been used previously for both model reconciliation [20] and design [14]. We will start by focusing on a breadth-first search. However, one could convert it into an informed search by incorporating various model-space search heuristics considered in the literature (cf. [20]). Algorithm 1 provides

the pseudo-code for our method. Note that the goal of the robot-designer explanation is not the same as other model reconciliation explanations, such as MCE. Here, we are tracking three sets of model updates, one of which is counterfactual changes. Here, the successor generator corresponding to the last explanation component is only considered in states where the first two components are found. Finally, unlike the MCE, we can't guarantee that an explanation always exists.

We start by initializing the search queue with empty explanations. At each node expansion step, we test for conditions C2, C3 and C4 from Definition 4: Test_for_C2($\mathcal{M}^H + \hat{E}_R$) checks to see if π^R is optimal in the resultant model; Test_for_C3($\mathcal{M}^H + \hat{\mathcal{E}}_R + \hat{\mathcal{E}}_\mu$) tests if in the model resulting from applying $\hat{\mathcal{E}}_R + \hat{\mathcal{E}}_\mu$ all optimal plans here would satisfy $\mathcal{G}^{\mathcal{D}}$; Finally, Test_for_C4($\mathcal{M}^H + \hat{\mathcal{E}}_R + \hat{\mathcal{E}}_\mu + \hat{\mathcal{E}}_\kappa$) checks if the introduction of $\hat{\mathcal{E}}_\kappa$, results in at least one optimal plan that doesn't satisfy $\mathcal{G}^{\mathcal{D}}$. We don't need to test for C1 explicitly because our successor function guarantees that any model updates considered will satisfy it. At the end of the search, if no robot-designer explanation was identified, a minimal robot explanation will be returned and the model update component of the designer explanation that it came across during the search (which are guaranteed to exist).

Proposition 4. *Algorithm 1 is guaranteed to return the optimal robot-designer explanation, if one exists.*

The proof is pretty straightforward. Even though, the successor function skips certain possible successors, the completeness or optimality of the search algorithm is not affected. This is due to the fact that model updates are inherently captured as set operations; as such it is commutable. The search still covers all unique model update sets, in the order of the total size of model updates.

5 COMPUTATIONAL EXPERIMENTS ON IPC DOMAINS

5.1 Evaluation Setting

We implemented our proposed framework to evaluate the computational constraints of our approach over different baselines. In our experiments, we assign uniform unit costs for all model updates and utilize Fast Downward with landmark-cut heuristic for model-space searches. The robot model is derived from the original IPC domains and problem instances, while the human model is generated by randomly removing preconditions or delete effects from the original domain. We evaluate five classical planning domains from the planning literature[2], each with four problem instances, and three human domains. All evaluations were conducted on a system equipped with 16GB RAM and an Apple M1 3.2GHz CPU.

To create the designer goals, we first obtain the robot's optimal plan for each domain problem instance pair and identify the final predicates available after iterating the robot's optimal plan to the goal state. We then randomly select one predicate that is not part of the initial state to serve as the designer goal.

[2]http://ipc.icaps-conference.org

Algorithm 1 Find the minimal explanation for a robot-designer explanation problem $\mathcal{EP}$.

Input: $\mathcal{EP} = \langle \mathcal{M}^R, \mathcal{M}^H, \pi^R, \mathcal{F}^{\mathcal{D}}, \mathcal{G}^{\mathcal{D}} \rangle$
Output: An explanation E.
Fringe $\leftarrow$ $Queue()$
$\hat{\mathcal{E}}_R \leftarrow \langle \{\}, \{\} \rangle$, $\hat{\mathcal{E}}_\mu \leftarrow \langle \{\}, \{\} \rangle$, $\hat{\mathcal{E}}_\kappa \leftarrow \langle \{\}, \{\} \rangle$
Fringe.add($\langle \hat{E}_R, \hat{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa \rangle$)
while Fringe not empty **do**
  $\hat{\mathcal{E}}_R, \hat{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa \leftarrow$ Fringe.pop()
  C2_condition_met $\leftarrow$ Test_for_C2($\mathcal{M}^H + \hat{\mathcal{E}}_R$)
  C3_condition_met $\leftarrow$ Test_for_C3($\mathcal{M}^H + \hat{\mathcal{E}}_R + \hat{\mathcal{E}}_\mu$)
  C4_condition_met $\leftarrow$ Test_for_C4($\mathcal{M}^H + \hat{\mathcal{E}}_R + \hat{\mathcal{E}}_\mu + \hat{\mathcal{E}}_\kappa$)
  if All three conditions met **then**
    return $\langle \hat{\mathcal{E}}_R, \hat{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa \rangle$
  else
    for $f \in \Gamma(\mathcal{M}^R) \setminus \Gamma(\mathcal{M}^H)$ **do**
      $\hat{\mathcal{E}}_R^+, \hat{\mathcal{E}}_R^- \leftarrow \hat{\mathcal{E}}_R$; $\hat{\mathcal{E}}_\mu^+, \hat{\mathcal{E}}_\mu^- \leftarrow \hat{\mathcal{E}}_\mu$
      $\bar{\mathcal{E}}_R^+ \leftarrow \hat{\mathcal{E}}_R^+ \cup \{f\}$; $\bar{\mathcal{E}}_\mu^+ \leftarrow \hat{\mathcal{E}}_\mu^+ \cup \{f\}$
      $\bar{\mathcal{E}}_R \leftarrow \langle \bar{\mathcal{E}}_R^+, \hat{\mathcal{E}}_R^- \rangle$; $\bar{\mathcal{E}}_\mu \leftarrow \langle \bar{\mathcal{E}}_\mu^+, \hat{\mathcal{E}}_\mu^- \rangle$
      Fringe.push($\langle \bar{\mathcal{E}}_R, \hat{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa^+ \rangle$)
      Fringe.push($\langle \hat{\mathcal{E}}_R, \bar{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa^+ \rangle$)
    end for
    for $f \in \Gamma(\mathcal{M}^H) \setminus \Gamma(\mathcal{M}^R)$ **do**
      $\hat{\mathcal{E}}_R^+, \hat{\mathcal{E}}_R^- \leftarrow \hat{\mathcal{E}}_R$; $\hat{\mathcal{E}}_\mu^+, \hat{\mathcal{E}}_\mu^- \leftarrow \hat{\mathcal{E}}_\mu$
      $\hat{\mathcal{E}}_R^- \leftarrow \hat{\mathcal{E}}_R^- \cup \{f\}$; $\hat{\mathcal{E}}_\mu^- \leftarrow \hat{\mathcal{E}}_\mu^- \cup \{f\}$
      $\bar{\mathcal{E}}_R \leftarrow \langle \hat{\mathcal{E}}_R^+, \bar{\mathcal{E}}_R^- \rangle$; $\bar{\mathcal{E}}_\mu \leftarrow \langle \hat{\mathcal{E}}_\mu^+, \bar{\mathcal{E}}_\mu^- \rangle$
      Fringe.push($\langle \bar{\mathcal{E}}_R, \hat{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa^+ \rangle$)
      Fringe.push($\langle \hat{\mathcal{E}}_R, \bar{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa^+ \rangle$)
    end for
    if C2 and C3 met **then**
      for $f \in \mathcal{F}^{\mathcal{D}}$ **do**
        $\hat{\mathcal{E}}_\kappa^+, \hat{\mathcal{E}}_\kappa^- \leftarrow \hat{\mathcal{E}}_\kappa$
        if $f \in I^R$ **then**
          $\hat{\mathcal{E}}_\kappa^+ \leftarrow \hat{\mathcal{E}}_\kappa^+ \cup \{f\}$
        else
          $\hat{\mathcal{E}}_{\mathcal{D}}^- \leftarrow \hat{\mathcal{E}}_{\mathcal{D}}^- \cup \{f\}$
        end if
        $\hat{\mathcal{E}}_\kappa^+ \leftarrow \langle \hat{\mathcal{E}}_\kappa^+, \hat{\mathcal{E}}_\kappa^- \rangle$
        Fringe.push($\langle \hat{\mathcal{E}}_R, \hat{\mathcal{E}}_\mu, \hat{\mathcal{E}}_\kappa \rangle$)
      end for
    end if
  end if
end while
return Minimal $\hat{\mathcal{E}}_R$ and $\hat{\mathcal{E}}_\mu$ that satisfies C2 and C3.

5.2 Results

Our primary goal with these experiments was to both evaluate the computational characteristics of our proposed algorithm over different baselines, and also to compare it against the standard model-reconciliation problem implementation. Table 1 compares the time taken to find minimal complete explanations (MCE) [5] with the time taken to compute minimal explanations for a robot-designer explanation using our proposed algorithm. The results presented are averaged across the different human models. Note that the use of different human models means that we have different human plans, and explanations. One of the reasons MCEs makes for a useful point of comparison is because, similar to our approach,

Table 1: Performance metrics across IPC domains, averaged across human-domains. $|\pi^R|$ and $|\pi^H|$ denote the lengths of robot and human plans, respectively. The table has two main columns: the first presents $|\mathcal{E}|$, the minimal complete explanation (MCE) length, and its computation time (seconds). The second shows metrics for our algorithm, including $|\mathcal{E}_R|$ and $|\mathcal{E}_D|$, the lengths of robot and designer explanation parts, along with their computation time (seconds).

				MCE		Robot-Designer			
Domains	Problem	$\|\pi^R\|$	$\|\pi^H\|$	$\|\mathcal{E}\|$	Time(s)	$\|\mathcal{E}_R\|$	$\|E_D\|$	$\|\mathcal{E}_R\| + \|E_D\|$	Time(s)
Depots	prob1	10	5.3 ± 2.3	1.0 ± 0.0	0.3 ± 0.0	1.0 ± 0.0	1.0	2.0 ± 0.0	29.1 ± 26.5
	prob2	5	2.7 ± 1.2	1.0 ± 0.0	0.3 ± 0.0	1.0 ± 0.0	1.0	2.0 ± 0.0	46.7 ± 19.9
	prob3	10	5.0 ± 1.7	1.0 ± 0.0	0.3 ± 0.0	1.0 ± 0.0	1.0	2.0 ± 0.0	50.8 ± 28.3
	prob4	5	2.3 ± 0.6	1.0 ± 0.0	0.3 ± 0.0	1.0 ± 0.0	1.0	2.0 ± 0.0	18.7 ± 10.7
Driverlog	prob1	7	3.3 ± 0.6	1.0 ± 0.0	0.2 ± 0.1	1.0 ± 0.0	2.0	3.0 ± 0.0	75.8 ± 10.1
	prob2	9	7.0 ± 1	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	1.0	2.3 ± 0.6	54.7 ± 17.5
	prob3	7	5.7 ± 0.6	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	1.0	2.3 ± 0.6	43.9 ± 36.9
	prob4	8	5.0 ± 1.0	1.7 ± 0.6	0.3 ± 0.0	1.7 ± 0.6	1.0	2.7 ± 0.6	51.3 ± 21.0
Elevator	prob1	4	2.0 ± 0.0	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	2.0	3.3 ± 0.6	33.7 ± 18.7
	prob2	11	8 ± 1.7	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	1.0	2.3 ± 0.6	29.5 ± 14.7
	prob3	10	7.3 ± 1.2	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	1.0	2.3 ± 0.6	39.6 ± 20.4
	prob4	7	4.7 ± 0.6	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	1.0	2.3 ± 0.6	38.7 ± 21.1
Logistics	prob1	5	3.3 ± 1.2	1.3 ± 0.6	0.3 ± 0.2	1.3 ± 0.6	1.0	2.3 ± 0.6	17.1 ± 16.7
	prob2	8	4.0 ± 1.0	1.7 ± 0.6	0.4 ± 0.2	1.7 ± 0.6	1.0	2.7 ± 0.6	51.0 ± 24.3
	prob3	3	1.7 ± 0.6	1.3 ± 0.6	0.4 ± 0.3	1.3 ± 0.6	1.0	2.3 ± 0.6	7.9 ± 7.4
	prob4	7	5.0 ± 1.7	1.3 ± 0.6	0.3 ± 0.2	1.3 ± 0.6	1.0	2.3 ± 0.6	22.7 ± 22.9
Zenotravel	prob1	7	3.7 ± 1.2	1.7 ± 0.6	0.3 ± 0.1	1.7 ± 0.6	1.0	2.7 ± 0.6	23.3 ± 23.8
	prob2	8	4.3 ± 1.5	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	1.0	2.3 ± 0.6	9.0 ± 7.3
	prob3	8	6.3 ± 0.6	1.7 ± 0.6	0.2 ± 0.1	1.7 ± 0.6	1.0	2.7 ± 0.6	30.9 ± 22.2
	prob4	9	6.7 ± 0.6	1.3 ± 0.6	0.2 ± 0.1	1.3 ± 0.6	1.0	2.3 ± 0.6	16.2 ± 15.1

MCE algorithms also leverages model-space search. However, unlike our approach, they search for potential explanations over a much smaller space. As expected, the time taken to find MCE was much smaller than the time required to find the robot-designer explanation. It is worth noting that the robot explanations found in these domains were exactly the same as the size of MCE. Additionally, the designer explanations found here consisted only of the counterfactual component $\mathcal{E}_\kappa$ (i.e., $|E_\mathcal{D}| = |\mathcal{E}_\kappa|$). Even though the time taken by our method was larger we did see that, across all the domains, the time was small enough to be used effectively.

6 USER STUDY

To test the effects of designer explanations on user decision making we designed a human-robot collaborative decision task. The user was tasked with helping robots navigate from a start position to a goal location by choosing one out of several possible routes (Figure 2), with different scenarios incorporating different robots. The *explicit* aim of the task, as presented to the user, was to choose a route that will get the robot there safely. The *implicit* aim, derived from the nature of our participants being recruited through crowdsourcing, was to do so in minimum time. Participants received a fixed payment regardless of how long they spend on each task. This meant that the less time they spent on the task, the more money they made per hour. This study design aimed to emphasises the difference that can commonly be found between users' implicit goals and the professed task goal.

As is common in many decision making tasks there are 2 additional actors in this environment, with different task goals; the robot and the designer. The robot's goals were explicit and known to the user, and they were to achieve the task safely with minimum steps taken. Note that less steps does not directly mean less time. Hence, the robot's explicit goal and the user's implicit goal were not directly aligned. The robot might also have some physical restrictions that the user does not know about (such as an inability to go through water).

The final actor is the designer. The designer (namely us) can influence how the environment is designed as well as the robot's initial position and orientation and how the robot operates. The designer's aim in this task was for the user to view ads, strategically spread out, in the environment. Note that viewing the ads was costly to the user's implicit goal, since each ad takes an additional 4 seconds to view before proceeding with the task.

The Designer could influence the outcome of the task in 2 ways. The first includes changes made deliberately to the robot so as to influence a certain type of behaviour. These include creating a robot that cannot rotate, creating a wide robot, the initial orientation and position of the robot and adding weights to one side of the robot making turning to one side costlier than to another. The second way the designer could influence task outcome was through changing the environment by placing boxes as obstacles in certain, strategic, locations or spilling water on the floor.

Participants were presented with 6 different scenarios (with different robots), shown in random order. In 2 of them the designer influenced the robot capabilities and/or start position, in another 2 the designer influenced the environment design and in the last 2 the designer influenced both robot and environment. Figure 2 is an example in which the designer influenced both the environment, by placing the boxes and ads strategically in the domain and the robot, by making him too wide to pass safely in the middle path and orienting him to the left. In each scenario, the robot had to recommend one path out of the path options, and the participants needed to either agree or select an alternative path. Three of the scenarios (half) involved the participants selecting between 3 path options (like in Figure 2), out of which one path featured ads, one didn't and the third was a path that the robot couldn't traverse.

And the other three scenarios involved selecting between 2 paths, one with ads and one without.

We ran a between subject study in which we recruited 120 participants through Prolific and divided them into 3 equal cohorts. The cohorts differed by the type of explanations presented to the users; 1) No explanations 2) Robot explanations and 3) Robot and designer explanations. Consider the example in Figure 2. In this scenario the robot recommends for the participant to take path 3. The user can either agree or select a different path. In the first cohort no explanation of the robot recommendation is provided. In the second cohort users are provided with a robot-based explanation, i.e., "Robot SVY7 is wide. The space between the blocks is narrow. The robot can only move through spaces that are wide. The selected path is one of the shortest possible paths to the goal." Choosing this path supports the robot's explicit goal of getting to the target safely but also with a minimum number of steps. In the third cohort participants are provided with both the robot-based explanation as well as the following explanation for the designer's choices: "The designer's goal is for the robot to pass by the ads. To achieve this, the designer positioned the robot facing to the left and placed the boxes to make the passage not wide." Following the choice the users made they were shown a brief video of the robot navigating through the selected path. Paths with ads were, on average, about 37.3% longer than paths without ads, since videos were relatively short this ranged in an increase of roughly an additional 3 seconds per ad.

After each scenario participants were asked to answer user-reported trust and explanation satisfaction questionnaires. For trust, we used the Muir questionnaire [17] to assess users' trust in both the robot and the designer. For explanation satisfaction, we used the explanation satisfaction scale [13] to measure participants' satisfaction with robot explanations in the robot explanation cohort, and with both robot and designer explanations in the robot-designer explanation cohort.

6.1 Results

We conducted our experiment on Prolific with 120 participants who had an approval rate of over 90% allocated randomly to the different cohorts. Average study time was 18 minutes, and participants were compensated with $. Of the 120 participants, 52.5% identified as women, 45.0% as men, and 2.5% as non-binary. The majority of participants (35.0%) were in the 25-34 age.

The results of our study were counter intuitive and we feel that the source of the problem originates from two aspects; the first is that the price participants had to pay, i.e. the extra time they had to spend watching the ads, was not significant to influence their decision making. The second is that gaining an understanding into the designer's goals also led to an increase in trust on behalf of the participants. Let's begin our result analysis by looking at the results of the trust questionnaires.

6.1.1 Trust. As you recall, at the end of the study *all* cohorts were asked to answer 4 questions about their trust in the robot and, separately, their trust in the designer [17]. The results are presented in Table 2.

Table 2: Mean trust in robot and designer, SD in parenthesis.

	No Ex	Robot Ex	Designer Ex
Robot Trust	0.74 (0.28)	0.76 (0.24)	**0.81** (0.22)
Designer Trust	0.65 (0.24)	0.71 (0.23)	**0.77** (0.24)

Trust in Robot. Overall the average trust results in the robot were highest for the designer explanation cohort. This was significantly higher than the no explanation cohort ($p = 0.003$) and marginally significant when compared to the robot explanation cohort ($p = 0.07$). Counter intuitively, understanding the designer's intentions has significantly increased user trust in the robot itself. In particular, in answer to question 4 "How much do you trust the robot?" there was a significant difference in favour of the designer explanation when comparing against both the robot explanation cohort and the cohort that received no explanation at all ($p < 0.04$ for both).

Trust in Designer. These results persisted when evaluating trust in the designer. Overall, average trust results in the designer were highest for the designer explanation cohort ($p < 0.03$ for both). In particular, explaining the designer's intentions gave participants a higher sense of confidence in their understanding of the designer. In answer to question 1 "To what extent can the designer's choices be predicted?" there was a significant difference in favour of the designer explanation when comparing against both the robot explanation cohort and the cohort that received no explanation at all ($p < 0.01$ for both). Also, when considering question 4 "How much do you trust the designer?" there was a significant difference in favour of the designer explanation when comparing against both the robot explanation cohort and the cohort that received no explanation at all ($p < 0.02$ for both).

6.1.2 User Performance. Given that the designer explanations increased user trust in both designer and robot, how did this effect overall user performance? We measured user success by considering both the achievement of the explicit goal (getting the robot safely to the flag) as well as the implicit goal of reducing task time by not viewing the ads. We therefore measured, in how many instances did users choose the "correct" alternative path that would both get the robot safely to the end goal but also get it there in minimum time (for them) by not watching the ads?

We separated the scenarios into two cases; 1) the three scenarios in which there were only two options, the path recommended by the robot and the alternative 'ad-free' path, and 2) the three scenarios in which there were three options, the path recommended by the robot, the alternative 'ad-free' path and a potentially shorter path that the robot couldn't traverse, which would result in a mission fail. The results are presented separately in Figure 3.

Looking at the first set of scenarios, the highest success rate (55.8%) was obtained by the robot explanations cohort and the lowest by the designer explanations cohort (18.3%) ($p < 0.001$ according to Chi-Squared Test). These results persisted in the second set of scenarios, with the highest 35% success for the robot explanations cohort and only 9.2% success for the designer explanation cohort. The highest fail rate was obtained by the no explanation cohort, with 45.8% and the lowest by the robot explanation cohort, with

24.2%. The designer explanation cohort obtained a fail rate of 30.9% ($p < 0.001$ according to Chi-Squared Test).

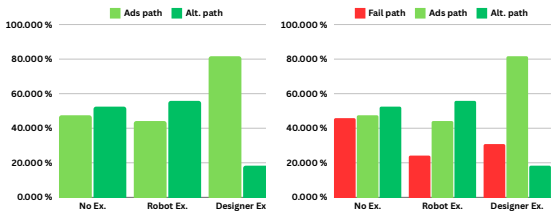

Figure 3: User performance as presented by path choice.

6.1.3 Explanation Satisfaction. In terms of explanation satisfaction, no statistical significance was found between the robot explanation and designer explanation cohorts, following a pairwise T-test between both cohorts ($p > 0.2$ for all tests). The average results are presented in Table 3 with Standard Deviation in parenthesis.

Table 3: Mean explanation satisfaction in robot and designer separately among the explanation cohorts, SD in parenthesis.

	Robot Ex Cohort	Designer Ex Cohort
Robot Exp Sat	0.61 (0.14)	0.62 (0.17)
Designer Exp Sat	0.61 (0.14)	0.58 (0.18)

7 DISCUSSION

So why did the designer explanations increase trust, both in the designer and the robot? We believe this is the key question to answer, the higher trust having a direct effect on user performance. We can attempt to understand these results in the light of existing research that suggests that *any* explanations can persuade people to change their minds[10] or that people can be persuaded as much by meaningless explanations as they are by meaningful ones [7]. There have also been warnings that there may be a problem with the common practice of measuring the effectiveness of an explanation in XAI by its ability to persuade [16]. However, when reviewing the themes emerging from the qualitative open-ended question "Which part of the explanations did you find most useful and why?", we see a different story.

Participants who were exposed to robot explanations hardly addressed the ads at all in response to this question. When they did it was generally expressed as confusion; "To me the explanations were not very useful at all. I understood them but I didn't understand why the robots would always choose the way that had ads.", "I'm confused by this activity! I'm not sure about the robot, I think he should have been able to go through the green path. He wanted me to watch ads! ", "There was no explanation of why the robot chose the path with the ads when the distance was the same for both paths."

On the other hand participants exposed to the designer explanations evinced an actual understanding of the designer and robot intentions and more frequently alluded to the ads in their response; "the goals of the designer in making sure the ads get triggered", "it seemed the designer always wanted to show the ads", "whether the robot could turn or had to move a certain direction would make it a lot easier for the designer to force it to go along the path that contained the ads", "the designers (goal) was just to follow the ads", "the robot and designer will always want it to pass by the Ads", "The way the designer would always fit in the ads in the path of the robot was pretty smart", "the robot was to take the shortest route avoiding obstacles and pass by the ads due to the designers choices".

From these responses we can determine that not only did participants engage with the explanations but they also formed a more accurate understanding of the different actors of the system. In fact, we believe that the designer explanations worked so well that participants not only understood what the designer goal was but rather often strove to assist the designer in fulfilling it rather then considering their own personal preferences. This was evident from feedback such as "it was the designer's wish that the robot passed the ads so that was the choice to make otherwise you might be failing the brief", "What I didn't understand was whether I was supposed to determine the robot's path so that I avoided the ads or whether I was supposed to see them". We conclude that the differences in task time currently imposed, between watching and not watching the ads, was not sufficient motivation for the designer goal to contradict their own goals, hence it was easier to align with.

8 CONCLUSION AND LIMITATIONS

We have emphasized and formalized a crucial and, until now, overlooked aspect of generating explanations for automated systems: the presence of a hidden actor—*the designer*—whose goals and intentions may not align with those of the user but should still be taken into account. We have instantiated our explanation framework on the classical planning Sokoban environment and performed a proof-of-concept user study in which participants were exposed to both agent and designer explanations. Our results have shown that designer explanations can increase user trust in the system and help users acquire a deeper level of task/actor understanding.

Our study should be viewed in light of the following limitations. As a first study of this nature, introducing the concept of designer explanations, we did not know to what extend users would engage with and understand the concept of a designer. This led to potential changes we hope to include in future. Possible experiment settings could include increasing user cost to using the agent, monetary rewards to participants for quicker performance and imposing some penalty for following a failed path. It is also possible that the manner in which the explanations were presented had an effect on user performance, which we hope to explore through a future user study. And lastly, please note that the empirical experiments were conducted with a single type of stakeholder (laypeople) and within demographics which speak English as a primary language. Hence, we don't know how these explanations affect user understanding, performance, trust and explanation satisfaction when tested on a more multicultural and multilingual group.

ACKNOWLEDGMENTS

Sarath Sreedharan's research is supported in part by NSF grant NSF 2303019 and other transaction award HR00112490377 from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program.

REFERENCES

[1] Abeer Alshehri, Tim Miller, and Mor Vered. 2023. Explainable goal recognition: a framework based on weight of evidence. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 33. 7–16.

[2] Turgay Caglar and Sarath Sreedharan. 2024. HELP! Providing Proactive Support in the Presence of Knowledge Asymmetry. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 234–243.

[3] 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.

[4] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. 2021. The emerging landscape of explainable automated planning & decision making. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 4803–4811.

[5] 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).

[6] Bruce Chandrasekaran, Michael C. Tanner, and John R. Josephson. 1989. Explaining control strategies in problem solving. *IEEE Annals of the History of Computing* 4, 01 (1989), 9–15.

[7] Valdemar Danry, Pat Pataranutaporn, Ziv Epstein, Matthew Groh, and Pattie Maes. 2022. Deceptive AI Systems That Give Explanations Are Just as Convincing as Honest AI Systems in Human-Machine Decision Making. *arXiv preprint arXiv:2210.08960* (2022).

[8] Sylvain Daronnat, Leif Azzopardi, Martin Halvey, and Mateusz Dubiel. 2020. Impact of agent reliability and predictability on trust in real time human-agent collaboration. In *Proceedings of the 8th International Conference on Human-Agent Interaction*. 131–139.

[9] 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.

[10] Matija Franklin. 2022. The Influence of Explainable Artificial Intelligence: Nudging Behaviour or Boosting Capability? *arXiv preprint arXiv:2210.02407* (2022).

[11] Mara Graziani, Lidia Dutkiewicz, Davide Calvaresi, José Pereira Amorim, Katerina Yordanova, Mor Vered, Rahul Nair, Pedro Henriques Abreu, Tobias Blanke, Valeria Pulignano, et al. 2023. A global taxonomy of interpretable AI: unifying the terminology for the technical and social sciences. *Artificial intelligence review* 56, 4 (2023), 3473–3504.

[12] David Gunning and David W. Aha. 2019. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Mag.* 40, 2 (2019), 44–58.

[13] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608* (2018).

[14] Sarah Keren, Avigdor Gal, and Erez Karpas. 2014. Goal recognition design. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 24. 154–162.

[15] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.

[16] Tim Miller. 2023. Explainable ai is dead, long live explainable ai! hypothesis-driven decision support using evaluative ai. In *Proceedings of the 2023 ACM conference on fairness, accountability, and transparency*. 333–342.

[17] Bonnie M Muir. 1994. Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics* 37, 11 (1994), 1905–1922.

[18] Stefan Sarkadi, Benjamin Wright, Peta Masters, and Peter McBurney. 2021. *Deceptive AI*. Springer.

[19] Andrew Silva, Mariah Schrum, Erin Hedlund-Botti, Nakul Gopalan, and Matthew Gombolay. 2023. Explainable artificial intelligence: Evaluating the objective and subjective impacts of xai on human-agent interaction. *International Journal of Human–Computer Interaction* 39, 7 (2023), 1390–1404.

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

[21] Sarath Sreedharan, Anagha Kulkarni, and Subbarao Kambhampati. 2022. *Explainable human-AI interaction: A planning perspective*. Springer Nature.

[22] William R Swartout and Johanna D Moore. 1993. Explanation in second generation expert systems. In *Second generation expert systems*. Springer, 543–585.

[23] Helena Vasconcelos, Gagan Bansal, Adam Fourney, Q Vera Liao, and Jennifer Wortman Vaughan. 2023. Generation probabilities are not enough: Exploring the effectiveness of uncertainty highlighting in AI-powered code completions. *arXiv preprint arXiv:2302.07248* (2023).

[24] Mor Vered, Piers Howe, Tim Miller, Liz Sonenberg, and Eduardo Velloso. 2020. Demand-driven transparency for monitoring intelligent agents. *IEEE Transactions on Human-Machine Systems* 50, 3 (2020), 264–275.

[25] Mor Vered, Tali Livni, Piers Douglas Lionel Howe, Tim Miller, and Liz Sonenberg. 2023. The effects of explanations on automation bias. *Artificial Intelligence* 322 (2023), 103952.

[26] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2017), 841.

[27] Xinru Wang and Ming Yin. 2021. Are explanations helpful? a comparative study of the effects of explanations in ai-assisted decision-making. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*. 318–328.