Foundations of Explanations as Model Reconciliation

Sarath Sreedharan[a,1,*], Tathagata Chakraborti[b,1], Subbarao Kambhampati[a]

[a] *Arizona State University*
[b] *IBM Research AI*

Abstract

Past work on plan explanations primarily involved the AI system explaining the correctness of its plan and the rationale for its decision in terms of its own model. Such soliloquy is wholly inadequate in most realistic scenarios where users have domain and task models that differ from that used by the AI system. We posit that the explanations are best studied in light of these differing models. In particular, we show how explanation can be seen as a "model reconciliation problem" (MRP), where the AI system in effect suggests changes to the user's mental model so as to make its plan be optimal with respect to that changed user model. We will study the properties of such explanations, present algorithms for automatically computing them, discuss relevant extensions to the basic framework, and evaluate the performance of the proposed algorithms both empirically and through controlled user studies.

Keywords: Explainable AI, Automated Planning, Mental Models

*Corresponding author.
URL: ssreedh3@asu.edu (Sarath Sreedharan)
[1] Equal contribution.

Contents

1. Introduction

1.1. Explanations cannot be a soliloquy!

There has been significant renewed interest recently in developing AI systems that can automatically provide explanations of their decisions to humans in the loop. While much of the interest has been focused on learning systems that can explain their classification decisions [1], a related broader problem involves providing explanations in the context of sequential human-AI interactions and human-in-the-loop decision making systems. [2]

In such scenarios, the automated agents are called upon to provide explanation of their behavior or plans, either in terms of the process that generated those plans, the domain knowledge they have been derived from, or ultimately their execution time considerations [3]. Although explanation of plans has been investigated in the past [4, 5, 6, 7] much of those works involved the planner explaining its decisions with respect to its own model (i.e. current state, actions, and goals) and assuming that this "*soliloquy*" also helps the human in the loop. While such a sanguine assumption may well be requited when the explainee is an expert debugger (e.g. the developer of the system) and is intimately familiar with the innards of the agent's model, it is completely unrealistic in most human-AI interaction scenarios. At the end of the day, the requirements for the content of an effective explanation depends largely on the nature of the end-user persona who is being explained to. [8, 9]

For end users, two considerations that show up for the explanation problem. The first is the computation power of the explainee – i.e. how comfortable are they with planning and what is the quality of plans they can compute given a planning problem. Most existing work, as described above, addresses this problem either explicitly or implicitly, often engaging the explainee in explanatory dialogue. [10, 11, 12] The other consideration that comes up, particularly when dealing with end users, is that humans in the loop may often have a domain and task model that differs significantly from that used by the AI system. This is particularly true when end users are not domain experts, and we will see later

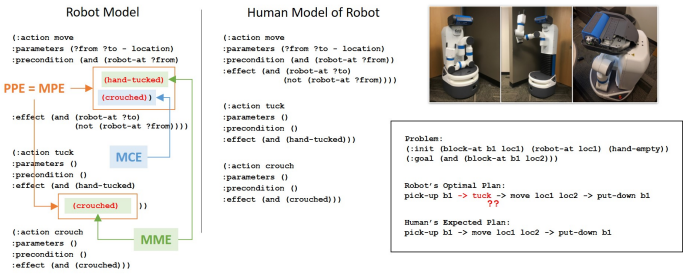


Figure 1: The agent uses it's model of the task $\mathcal{M}_h^R$ to come up with a plan $\pi^*_{\mathcal{M}^R}$, while a human observer makes use of their estimation to the robot's model $\mathcal{M}_h^R$ to make sense of their plan. An explanation for $\pi^*_{\mathcal{M}^R}$ here consists of an update to this mental model $\mathcal{M}_h^R$ of the human so that the decision made by the agent is the best in the updated mental model $\widehat{\mathcal{M}}_h^R$.

in Section 7.1, many applications that have to deal with differences between the AI system and the mental model of the user.

We posit then that the need for plan explanations should be understood in the context of the explainee's misunderstandings of the robot model – i.e. differences in the agent model $\mathcal{M}^R$ and the mental model $\mathcal{M}_h^R$ of the user – and consequently the explanation process can be seen as an agent's attempt to move its understanding of the user's mental model – $\hat{\mathcal{M}}_h^R$ – to be in conformance with its own. This process of *model reconciliation* thus forms the core of the explainable AI problem.[2] This is visualized in Figure 1.

1.2. Contributions and Paper Outline

In this paper, we provide a comprehensive treatise of this model reconciliation process, from its fundamental properties to a series of extensions that makes it amenable to real world applications.

Section 2 We start with the basics of explanations as a process of reconciliation of models. Specifically, we emphasize how the mental model of the explainee must form a core component of the explanation process.

[2]Though we primarily focus on the automated planning problem in this paper, it should become clear pretty soon that the concept of model reconciliation is generally applicable to different models of decision-making in AI.

Section 2.1 To this end, we will show how the task of generating explanations becomes one of reasoning in the space of differences between the model of the explainer and the mental model of the explainee.

Section 2.2 We provide a detailed analysis of the properties of various explanation types that can be generated in this framework.

Section 3 We will then relax some of the restrictive assumptions regarding the mental model made in the basic formulation of the model reconciliation problem and show how they can be dealt with under the same framework. This includes the ability to deal with incomplete information about the mental model (Section 3.1), multiple explainees in the loop (Section 3.2), as well as a few pointers (Section 3.3) to other extensions of the same framework that deal with expertise level, unsolvability, and differences in representation and computational power of the mental model.

Section 4-5 We will also evaluate these algorithms extensively in terms of their empirical properties on a few benchmark planning domains as well as through controlled user studies in a mock search and reconnaissance domain.

As we discuss in Section 6, to the best of our knowledge, the model reconciliation framework for explanations is the only approach in existing literature that touches upon the three key properties – social, selective, and contrastive – of explanations outlined in [13] (a more detailed discussion of the properties are provided in Section 6). Finally, in Section 7.1, we will end with a brief discussion on how some of these approaches have been adopted for explainable decision making systems in the real world.

1.3. Running Examples

We will be using the following two domains as running examples throughout the rest of the paper. The first one is a toy example that will be used to illustrate the salient aspects of different kinds of explanations that come out of the model reconciliation framework, while the latter (mimicking a real world domain) will be used later to evaluate these concepts with end users.

1.3.1. The Fetch Domain

Consider the Fetch robot whose design requires it to `tuck` its arms and lower its torso or `crouch` before moving. This is not obvious to a human navigating it and it may lead to an unbalanced base and toppling of the robot if the human deems such actions as unnecessary. The move action for the robot is described in PDDL [14] (the models are formally defined in Section 2) in the following model snippet –

```
(:action move
:parameters     (?from ?to - location)
:precondition   (and (robot-at ?from) (hand-tucked) (crouched))
:effect         (and (robot-at ?to) (not (robot-at ?from))))

(:action tuck
:parameters     ()
:precondition   ()
:effect         (and (hand-tucked) (crouched)))

(:action crouch
:parameters     ()
:precondition   ()
:effect         (and (crouched)))
```

Notice that the `tuck` action also involves a lowering of torso so that the arm can rest on the base once it is tucked in.[3] Now, consider a planning problem where the the robot needs to transport a block from one location to another, with the following initial and goal states –

```
(:init (block-at b1 loc1) (robot-at loc1) (hand-empty))
(:goal (and (block-at b1 loc2)))
```

An optimal plan for the robot involves a `tuck` action followed by a `move`:

[3]Fetch User Manual: `https://docs.fetchrobotics.com/`

Figure 2: The Fetch in the crouched position with arm tucked (left), torso raised and arm outstretched (middle) and the rather tragic consequences of a mistaken action model (right showing a fractured head from an accident).

```
pick-up b1 -> tuck -> move loc1 loc2 -> put-down b1
```

The human, on the other hand, expects a much simpler model, as shown below.[4] In the human's model of the robot, `move` action does not have the preconditions for tucking the arm and lowering the torso, and `tuck` does not automatically lower the torso either. This means the behavior expected by the human may not match what is generated by the robot.

```
(:action move
:parameters     (?from ?to - location)
:precondition   (and (robot-at ?from)
:effect         (and (robot-at ?to) (not (robot-at ?from))))

(:action tuck
:parameters     ()
:precondition   ()
:effect         (and (hand-tucked))
```

[4]This is actually a common problem with deploying any software to end users: generic user models are used to model the average user and these lack details and nuances of the system at hand that only experts would be aware of.

```
(:action crouch
:parameters     ()
:precondition   ()
:effect         (and (crouched)))
```

The original plan is no longer optimal to the human who can envisage better alternatives (a shorter plan without the extra `tuck` action) in their mental model. An explanation here is a model update that can address this disagreement.

```
Explanation >> MOVE_LOC1_LOC2-has-precondition-HAND-TUCKED
```

This correction brings the mental model (i.e. the model the human believes is being used by the robot) closer to the robot's ground truth and is necessary and sufficient to make the robot's plan optimal in the resultant domain so that the human cannot envisage any better alternatives. This process of selective update of human mental model to clarify the status of the current plan forms the essence of the model reconciliation process.

1.3.2. The Urban Search and Reconnaissance Domain

The second domain is a typical Urban Search and Reconnaissance (USAR) domain[5] [15, 16] where a remote robot is put into disaster response operation often controlled partly or fully by an external human commander, as shown in Figure 3. The robot's job is to scout areas that may be otherwise harmful to humans and report on its surroundings as instructed by the external supervisor. The scenario can also have other internal agents (humans or robots) with whom the robot needs to coordinate. The USAR domain thus affords a rich set of characteristics, such as multiple agents distributed in space, partial observability, evolving domain models, and so on. The USAR domain is also ideal for visualizing to non-expert participants in comparison to, for example, logistics-type domains which should ideally be evaluated by experts. This became an important factor

[5] Video demonstrations of all examples in this domain can be viewed at https://ibm.box.com/v/aij-model-reconciliation. (Duration 5:52)

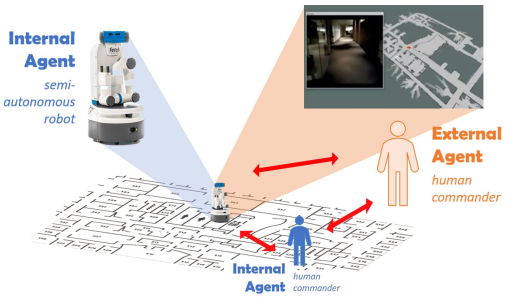


Figure 3: A typical USAR domain with an internal robot and an external commander.

while designing the user studies. The USAR domain is thus at once close to motion planning as easily interpreted by non-experts but also incorporates typical aspects of task plans such as preconditions and effects in terms of rubble removal, collapsed halls, etc. and relevant abilities of the robot. As such, simulated USAR scenarios provide an ideal testbed [15, 17, 18] for developing algorithms for effective human-robot interaction.

Here, even though all agents start off with the same model – i.e. the blueprint of the building – their models diverge as the internal agent interacts with the scene. Due to the disaster new paths may have opened up due to collapsed walls or old paths may no longer be available due to rubble. This means that plans that are valid and optimal in the robot's model may not make sense to the external commander. In the scenario in Figure 4, the robot is tasked to go from its current location marked blue to conduct reconnaissance in the location marked orange. The green path is most optimal in its current model but this is blocked in the externals mental model while the expected plan in the mental model is no longer possible due to rubble. Without removing rubble in the blocked paths, the robot can instead communicate that the path at the bottom is no longer blocked. This explanation preserves the validity and optimality of its plan in the updated model (even though further differences exist).

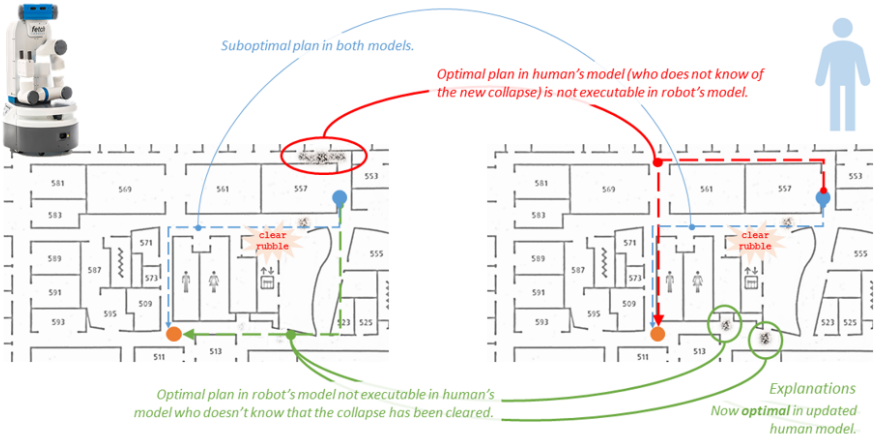


Figure 4: Model differences in the USAR domain.

2. Explanation as Model Reconciliation

Definition: **A Planning Problem** is a tuple $\mathcal{M} = \langle \mathcal{D}, \mathcal{I}, \mathcal{G} \rangle$ with domain $\mathcal{D} = \langle F, A \rangle$ – where F is a finite set of fluents that define a state $s \subseteq F$, and A is a finite set of actions – and initial and goal states $\mathcal{I}, \mathcal{G} \subseteq F$. Action $a \in A$ is a tuple $\langle c_a, \mathit{pre}(a), \mathit{eff}^{\pm}(a) \rangle$ where c_a is the cost, and $\mathit{pre}(a), \mathit{eff}^{\pm}(a) \subseteq F$ are the preconditions and add/delete effects, i.e. $\delta_{\mathcal{M}}(s, a) \models \perp$ *if* $s \not\models \mathit{pre}(a)$; *else* $\delta_{\mathcal{M}}(s, a) \models s \cup \mathit{eff}^{+}(a) \setminus \mathit{eff}^{-}(a)$ where $\delta_{\mathcal{M}}(\cdot)$ is the transition function.

This forms the classical definition of a planning problem [19] whose models are represented in the syntax of `PDDL` [14]. The solution to the planning problem is a sequence of actions or a (satisficing) *plan* $\pi = \langle a_1, a_2, \ldots, a_n \rangle$ such that $\delta_{\mathcal{M}}(\mathcal{I}, \pi) \models \mathcal{G}$. The cost of a plan π is given by $C(\pi, \mathcal{M}) = \sum_{a \in \pi} c_a$ if $\delta_{\mathcal{M}}(\mathcal{I}, \pi) \models \mathcal{G}$; ∞ otherwise. The cheapest plan $\pi^* = \arg\min_{\pi} C(\pi, \mathcal{M})$ is the (cost) optimal plan. We refer to the cost of the optimal plan in the model $\mathcal{M}$ as $C^*_{\mathcal{M}}$.

The model reconciliation framework introduces the *mental model* of the human in the loop into a planner's deliberative process, in addition to the planner's own model in the classical sense. As described previously, even if the robot is doing the best it can a plan π that is optimal in the robot's model may not be optimal in the human mental model and thus *inexplicable* from the

point of view of the human – this means that the human can come up with "better solutions" (in their mental model) to the planning problem at hand. The explanation process thus begins with the following question:

Q_1*: Why plan* π*?*

An explanation here needs to ensure that plan π that both the explainer and the explainee agree that this is the best decision that could have been made in the given problem. This can be achieved by the agent providing model artifacts to the explainee so that π is now also optimal in the updated mental model (we will refer to this as the completeness property later).

Definition **The Model Reconciliation Problem (MRP)** is the tuple $\langle \pi^*, \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle \rangle$ where $C(\pi^*, \mathcal{M}^R) = C^*_{\mathcal{M}^R}$, i.e. the plan π^* is the optimal plan in the robot model $\mathcal{M}^R$ but may not be so in the human mental model $\mathcal{M}_h^R$.

It is important to note here that $\mathcal{M}_h^R$ is the robot's approximation of the *information content* of the mental model – there is, of course, no PDDL inside the human's head (c.f. previous examples in Section 1.3). This mental model here is just a copy of the agent's own decision making problem (here, a planning problem but it can be any other model of decision making or even a graph) that the agent believes is held by the human. This model is thus a generative model of user expectations of the agent. Also note, these two models could pretty much differ along any aspects, including the initial state, goal, action definitions (including cost) and even fluents used. For notational convenience, we will assume there is a one-to-one correspondence between actions in the models $\mathcal{M}^R$ and $\mathcal{M}_h^R$. Though, we can easily use this to capture cases where one model have actions absent from the other, by assuming the other model has a dummy version the same action with unachievable preconditions.

Definition **An Explanation** is a solution to the model reconciliation problem in the form of (1) a model update $\mathcal{E}$ such that the (2) robot optimal plan is (3) also optimal in the update mental model.

(1) $\widehat{\mathcal{M}}_h^R \longleftarrow \mathcal{M}_h^R + \mathcal{E}$; and

(2) $C(\pi, \mathcal{M}^R) = C^*_{\mathcal{M}^R}$;

(3) $C(\pi, \widehat{\mathcal{M}}_h^R) = C^*_{\widehat{\mathcal{M}}_h^R}$.

The above question can also be posed in the following, more restricted, form:

Q_2: Why not a different plan $\hat{\pi}$?

Here $\hat{\pi}$ – an alternative solution proposed by the explainee – is referred to as a *foil*. [13] As before, the purpose of an explanation is to negate the foil so that both the human and the robot can come to the same page with regards to the decision that it has made, i.e. the explainee agrees that π is better than $\hat{\pi}$.

(4) $\delta_{\widehat{\mathcal{M}}_h^R}(\widehat{\mathcal{I}}_h^R, \pi) \models \widehat{\mathcal{G}}_h^R \ \wedge \ C(\pi, \widehat{\mathcal{M}}_h^R) < C(\hat{\pi}, \widehat{\mathcal{M}}_h^R)$.[6]

Note that Q_1, in essence, involves an implicit quantifier over all possible foils, as handled by Condition (3). This is thus a more conservative target and subsumes the case of the explicit foil – we will thus focus on this case more in this paper.[7] As we will explore in more detail later, this ability to use explanations as a medium of comparing solutions and foils is called the contrastive property, which forms a critical component of the explanation process.

Note that we only consider cases where the robot is explaining a decision it has made with respect to its model – the robot model need not be the ground truth. However, the robot can only explain with respect to what it believes to

[6] Note that the "closeness" or distance to an expected plan is modeled here in terms of cost optimality, but in general this can be any preference metric like plan similarity as investigated in existing literature on explicable planning [20, 21, 22] and plan similarity. [23, 24] This does not effect the algorithms described in this paper, since the computation of similarity is only invoked during the evaluation process of a particular node and the stopping criterion of the search, rather than the search process itself.

[7] An assumption we made here is that the computation power (or planning capability) of the human is the same as that of the planner, i.e. the human can compute the optimal plan given a planning problem. This assumption can be relaxed by requiring $|C(\pi, \widehat{\mathcal{M}}_h^R) - C^*_{\widehat{\mathcal{M}}_h^R}| < \delta$, to model an ϵ-optimal human or consider top-K plans [25] for hypothesis generation.

be true. The grander scope of explanatory dialogue may involve cases where it is wrong in its understanding of the model of the world (or is suboptimal) and thus needs to update its own model (or plan, respectively) iteratively in the course of further explanatory dialogue with the human in the loop, for example, in a decision support setting. [11, 26]

2.1. Model Space

In order to perform the model reconciliation process, we define the following state representation over planning problems –

$$\begin{aligned}\mathcal{F} = \; & \{\textit{init-has-f} \mid f \in F_h^R \cup F^R\} \cup \{\textit{goal-has-f} \mid f \in F_h^R \cup F^R\} \\ & \bigcup_{a \in A_h^R \cup A^R} \{\textit{a-has-precondition-f}, \textit{a-has-add-effect-f}, \\ & \qquad\qquad \textit{a-has-del-effect-f} \mid f \in F_h^R \cup F^R\} \\ & \cup \{\textit{a-has-cost-}c_a \mid a \in A_h^R\} \cup \{\textit{a-has-cost-}c_a \mid a \in A^R\}.\end{aligned}$$

A mapping function $\Gamma : \mathcal{M} \mapsto s$ represents any planning problem $\mathcal{M} = \langle\langle F, A\rangle, \mathcal{I}, \mathcal{G}\rangle$ as a state $s \subseteq \mathcal{F}$ as follows –

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

$$\begin{aligned}\Gamma(\mathcal{M}) = \; & \big\{\tau(f) \mid f \in \mathcal{I} \cup \mathcal{G} \cup \\ & \bigcup_{a \in A} \{f' \mid f' \in \{c_a\} \cup \textit{pre}(a) \cup \textit{eff}^+(a) \cup \textit{eff}^-(a)\}\big\}\end{aligned}$$

We can now define a *model-space search problem* $\langle\langle \mathcal{F}, \Lambda\rangle, \Gamma(\mathcal{M}_1), \Gamma(\mathcal{M}_2)\rangle$ with a new action set Λ containing unit model change actions $\lambda : \mathcal{F} \to \mathcal{F}$. The new transition or edit function is given by $\delta_{\mathcal{M}_1, \mathcal{M}_2}(s_1, \lambda) = s_2$ such that

`condition a` : $s_2 \setminus s_1 \subseteq \Gamma(\mathcal{M}_2)$, `condition b` : $s_1 \setminus s_2 \not\subseteq \Gamma(\mathcal{M}_2)$ and `condition c` $|s_1 \Delta s_2| = 1$ (Δ being the symmetric difference) are satisfied. This means that model change actions can only make a single change to a model at a time (starting from $\mathcal{M}_1$), and *all these changes are consistent with the target model* $\mathcal{M}_2$. The solution to a model-space search problem is given by a *set* of edit functions $\{\lambda_i\}$ that transforms the model $\mathcal{M}_1$ to $\mathcal{M}_2$, i.e. $\delta_{\mathcal{M}_1,\mathcal{M}_2}(\Gamma(\mathcal{M}_1), \{\lambda_i\}) = \Gamma(\mathcal{M}_2)$. An explanation can thus be cast as a solution to the model-space search problem $\langle\langle \mathcal{F}, \Lambda\rangle, \Gamma(\mathcal{M}_h^R), \Gamma(\widehat{\mathcal{M}})\rangle$ with the transition function $\delta_{\mathcal{M}_h^R,\mathcal{M}^R}$ such that Condition (3) above is preserved.[8]

2.2. Types of Explanations

Before we go on to different types of explanations, we consider the following requirements that characterize explanations in this multi-model setting.

R1. **Completeness -** Explanations of a plan should allow them to be compared and contrasted against other alternatives, so that no better solution exists. We enforce this property by requiring that in the updated human mental model the plan being explained is now optimal.

   – *An explanation is complete iff* $C(\pi, \widehat{\mathcal{M}}_h^R) = C^*_{\widehat{\mathcal{M}}_h^R}$.

R2. **Conciseness -** Explanation should be concise so that they are easily understandable to the explainee. The larger the explanation, the harder it is for the human to process that information. Thus the length of the explanation, i.e. the number of edits made as part of the explanation ($|\Gamma(\mathcal{M}_h^R)\Delta\Gamma(\widehat{\mathcal{M}}_h^R)|$), serves as a useful proxy or first approximation for the complexity of an explanation.

[8] Notice that we insisted that explanations must be compatible with the planner's model ($\mathcal{M}^2$ in the above definition). If this requirement is relaxed, it allows the planner to generate "explanations" that it knows are not true, and thus deceive the human. [27] While endowing the planner with such abilities may warrant significant ethical concerns, we note that the notion of white lies, and especially the relationship between explanations, excuses and lies has received very little attention and affords a rich set of exciting research problems. [28, 29]

Explanation Type	R1	R2	R3	R4
Plan Patch Explanation / VAL	✗	✓	✗	✓
Model Patch Explanation	✓	✗	✓	✓
Minimally Complete Explanation	✓	✓	✗	?
Minimally Monotonic Explanation	✓	✓	✓	?
(Approximate) Minimally Complete Explanation	✗	✓	✗	✓

Table 1: Requirements for different types of explanations.

R3. **Monotonicity -** This ensures that remaining model differences cannot change the completeness of an explanation, i.e. all aspects of the model that engendered the plan have been reconciled. Thus, monotonicity of an explanation subsumes completeness and requires more detail.

- *An explanation is monotonic iff*
  $C(\pi^*, \hat{\mathcal{M}}) = C^*_{\hat{\mathcal{M}}}\ \forall \hat{\mathcal{M}} : \Gamma(\widehat{\mathcal{M}})\Delta\Gamma(\mathcal{M}_h^R) \subset \Gamma(\hat{\mathcal{M}})\Delta\Gamma(\mathcal{M}_h^R)$.

That is no additional information revealed about the model should cause the human to question the validity of previous explanations. This is a very useful property to have. Doctors, for example, reveal different amount of detail to their patients as opposed to their peers, and often need to maintain monotonicity and resolve conflict of information [30] during the course of treatment. Further, the idea of completeness, i.e. withholding information on other model changes as long as they explain the observed plan, is also quite prevalent in how we deal with similar scenarios ourselves - e.g. progressing from Newtonian physics in high school to Einsteins Laws of Relativity in college.

R4. **Computability -** While conciseness deals with how easy it is for the explainee to understand an explanation, computability measures the ease of computing the explanation from the point of view of the planner.

We will now introduce different kinds of multi-model explanations that can participate in the model reconciliation process, propose algorithms to compute

them, and compare and contrast their respective properties. We note that the requirements outlined above are in fact often at odds with each other - an explanation that is very easy to compute may be very hard to comprehend. This (as seen in Table 1) will become clearer in course of this discussion.

A simple way to explain would be to provide the model differences pertaining to only the actions that are present in the plan being explained –

Definition: **A Plan Patch Explanation (PPE)** is given by –

$$\mathcal{E}^{PPE} = \Delta_{\{\mathcal{M}^R, \mathcal{M}_h^R\}} \bigcup_{f \in \{c_a\} \cup pre(a) \cup eff^+(a) \cup eff^-(a) : a \in \pi} \tau(f)$$

Clearly, such an explanation is easy to compute and concise by focusing only on plan being explained. However, it may also contain information that need not have been revealed, while at the same time ignoring model differences elsewhere in $\mathcal{M}_h^R$ that could have contributed to the plan being suboptimal in it. Thus, it is incomplete. One could adapt VAL [31, 32], to the multi-model setting to generate a version of PPE. VAL is plan validation tool which can simulate the execution of a plan in a given model. A multi-model VAL would need to extend this simulation to multiple models and compare and contrast the differing results of execution in the different models. Unfortunately, this would still suffer from the same limitations mentioned above. On the other hand, an easy way to compute a complete explanation would be to provide the entire model difference to the human –

Definition: **A Model Patch Explanation (MPE)** is given by –

$$\mathcal{E}^{MPE} = \Gamma(\mathcal{M}^R) \Delta \Gamma(\mathcal{M}_h^R)$$

This is also easy to compute but can be quite large and is hence far from being concise. Thus, in the following, we will try to minimize the size (and hence increase the comprehensibility) of explanations by searching in the space of models and thereby not exposing information that is not relevant to the plan being explained while still trying to satisfy as many requirements as we can.

Definition: **A Minimally Complete Explanation (MCE)** is the shortest possible explanation that is complete –

$$\mathcal{E}^{MCE} = \arg\min_{\mathcal{E}} |\Gamma(\widehat{\mathcal{M}})\Delta\Gamma(\mathcal{M}_h^R)| \text{ with R1}$$

The explanation provided before in the Fetch domain, as well as the one in the USAR domain (Section 1.3), are indeed the smallest domain changes that may be made to make the given plan optimal in the updated action model, and is thus an example of a minimally complete explanation.

The optimality criterion happens to be relevant to both the cases where the human expectation is better, or worse, than the plan computed by the planner. This might be counter to intuition, since in the latter case one might expect that just establishing feasibility of a better plan would be enough. Unfortunately, this is not the case, as can be easily seen by creating counter-examples where other faulty parts of the human model might disprove the optimality of the plan.

Proposition 1 – If $C(\pi^*, \mathcal{M}_h^R) < \min_\pi C(\pi, \mathcal{M}_h^R)$, then ensuring feasibility of the plan in the modified planning problem, i.e. $\delta_{\widehat{\mathcal{M}}}(\widehat{\mathcal{I}}, \pi^*) \models \widehat{\mathcal{G}}$, is a necessary but *not* a sufficient condition for $\widehat{\mathcal{M}} = \langle \widehat{D}, \widehat{\mathcal{I}}, \widehat{\mathcal{G}} \rangle$ to yield a valid explanation.

Note that a minimally complete explanation can be rendered invalid given further updates to the model. This can be easily demonstrated in our running example in the Fetch domain. Imagine that if, at some point, the human were to find out that the action move also has a precondition `(crouched)`, then the previous robot plan will no longer make sense to the human since now, according to the human's faulty model (being unaware that the tucking action also lowers the robot's torso) the robot would need to do *both* `tuck` and `crouch` actions before moving. Consider the following explanation in the Fetch domain instead –

```
Explanation >> TUCK-has-add-effect-CROUCHED
Explanation >> MOVE_LOC2_LOC1-has-precondition-CROUCHED
```

This explanation does not reveal all model differences but at the same time ensures that the plan remains optimal for this problem, irrespective of any other

changes to the model, by accounting for all the relevant parts of the model that engendered the plan. It is also the smallest possible among all such explanations.

Definition: **A Minimally Monotonic Explanation (MME)** is the shortest explanation that preserves both completeness and monotonicity –

$$\mathcal{E}^{MME} = \arg\min_{\mathcal{E}} |\Gamma(\widehat{\mathcal{M}})\Delta\Gamma(\mathcal{M}_h^R)| \text{ with R1 \& R3}$$

An MCE or MME solution may not be unique to an MRP problem. This can happen when there are multiple model differences supporting the same causal links in the plan - a minimal explanation can get by (i.e. guarantee optimality in the modified model) by only exposing one of them to the human. Interestingly, we showed in [33] how theoretically equivalent explanations are, in fact, sometimes interpreted differently by the explainee. The results from that study indicated a preference for explanations related to the effects of actions.

Proposition 2 – MCEs and MMEs are not unique, i.e. there might be multiple minimally complete and monotonic solutions to a given MRP.

Even though MCEs are an abridged version of an MME, it is easy to see that an MCE may not necessarily be part of an actual MME. This is due to the non-uniqueness property of MCEs and MMEs. This is illustrated in Figure 5.

Proposition 3 – An MCE may not be a subset of an MME, but it is always smaller or equal in size, i.e. $|\mathcal{E}^{MCE}| \leq |\mathcal{E}^{MME}|$.

2.3. Model Space Search for Minimal Explanations

In the following, we will see how the state space designed in Section 2.1 can be used in model-space search for computing MCEs and MMEs (computation of PPE and MPE follows directly from $\mathcal{M}^R$, $\mathcal{M}_h^R$ and π^*).

2.3.1. Model Space Search for MCEs

To compute MCEs, we employ A* search, similar to [34, 35], in the space of models, as shown in Algorithm 1. The algorithm is referred to as `MEGA` –

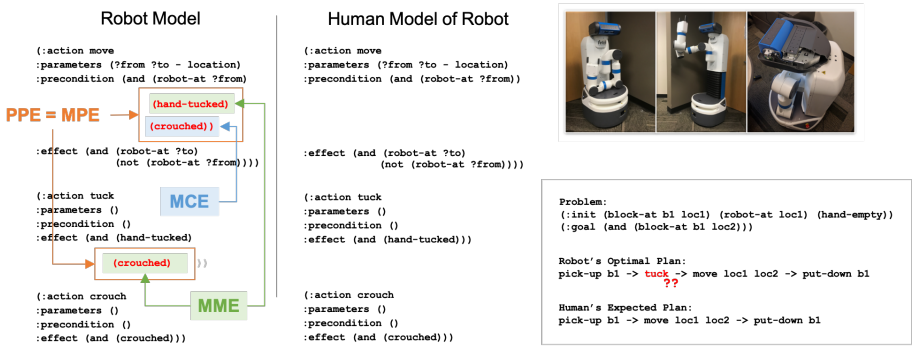


Figure 5: Illustration of the different kinds of explanations in the Fetch domain. Here the PPE and MPE are equivalent and involves notifying the human about both missing preconditions of the move action and the missing effect for the tuck action (which is the worst case for the former) and both longer than the MCE or the MME. Also, the MCE (which involves just notifying the human that there hand being tucked is a precondition for move action) is shorter than, and interestingly not a subset of, the MME.

Multi-model **E**xplanation **G**eneration **A**lgorithm. Given an MRP, we start off with the initial state $\Gamma(\mathcal{M}_h^R)$ derived from the human's expectation of a given planning problem $\mathcal{M}^R$, and modify it incrementally until we arrive at a planning problem $\widehat{\mathcal{M}}$ with $C(\pi^*, \widehat{\mathcal{M}}) = C^*_{\widehat{\mathcal{M}}}$, i.e. the given plan is explained. Note that the model changes are represented as a set, i.e. there is no sequentiality in the search problem. Also, we assign equal importance to all model corrections. We can easily capture differential importance of model updates by attaching costs to the edit actions λ - the algorithm remains unchanged. We also employ a selection strategy for successor nodes to speed up search (by overloading the way the priority queue is popped) by first processing model changes that are relevant to actions in π^*_R and π_H before the rest.

Proposition 4 – The successor selection strategy outlined in Algorithm 1 yields an admissible heuristic for model space search for minimally complete explanations.

Proof. Let $\mathcal{E}$ be the MCE for an MRP problem and let $\mathcal{E}'$ be any intermediate explanation found by our search such that $\mathcal{E}' \subset \mathcal{E}$, then the set $\mathcal{E} \setminus \mathcal{E}'$ must contain at least one λ related to actions in the set $\{a \mid a \in \pi^*_R \vee a \in \pi'\}$ (where

π' is the optimal plan for the model $\hat{\mathcal{M}}$ where $\delta_{\mathcal{M}_h^R,\mathcal{M}^R}(\Gamma(\mathcal{M}_h^R),\mathcal{E}') = \Gamma(\hat{\mathcal{M}})$. To see why this is true, consider an $\mathcal{E}'$ where $|\mathcal{E}'| = |\mathcal{E}| - 1$. If the action in $\mathcal{E} \setminus \mathcal{E}'$ does not belong to either π_R^* or π' then it can not improve the cost of π_R^* in comparison to π' and hence $\mathcal{E}$ can not be the MCE. Similarly we can show that this relation will hold for any size of $\mathcal{E}'$. We can leverage this knowledge about $\mathcal{E} \setminus \mathcal{E}'$ to create an admissible heuristic that considers only relevant changes.

2.3.2. Model Space Search for MMEs

As per the definition of MMEs, beyond the model obtained from the minimally monotonic explanation, there do not exist any models which are not explanations of the same MRP, while at the same time making as few changes to the original problem as possible. It follows that this is the largest set of changes that can be done on $\mathcal{M}^R$ and still find a model $\widehat{\mathcal{M}}$ where $C(\pi^*,\widehat{\mathcal{M}}) = C^*_{\widehat{\mathcal{M}}}$ - we are going to use this property in the search for MMEs.

Proposition 5 – $\mathcal{E}^{MME} = \arg\max_{\mathcal{E}} |\Gamma(\widehat{\mathcal{M}})\Delta\Gamma(\mathcal{M}^R)|$ such that $\forall\hat{\mathcal{M}}\, \Gamma(\hat{\mathcal{M}})\Delta\Gamma(\mathcal{M}^R) \subseteq \Gamma(\widehat{\mathcal{M}})\Delta\Gamma(\mathcal{M}^R)$ it is guarantee to have $C(\pi^*,\hat{\mathcal{M}}) = C^*_{\hat{\mathcal{M}}}$.

This is similar to the model-space search for MCEs, but this time starting from the robot's model $\mathcal{M}^R$ instead. The goal here is to find the largest set of model changes for which the explicability criterion becomes invalid for the first time (due to either suboptimality or inexecutability). This requires a search over the entire model space (Algorithm 2). We can leverage Proposition 3 to reduce our search space. Starting from $\mathcal{M}^R$, given a set of model changes $\mathcal{E}$ where $\delta_{\mathcal{M}_R,\mathcal{M}_H}(\Gamma(\mathcal{M}^R),\mathcal{E}) = \Gamma(\widehat{\mathcal{M}})$ and $C(\pi^*,\widehat{\mathcal{M}}) > C^*_{\widehat{\mathcal{M}}}$, no superset of $\mathcal{E}$ can lead to an MME solution. In Algorithm 2, we keep track of such unhelpful model changes in the list h_list. The variable $\mathcal{E}^{MME}$ keeps track of the current best list of model changes. Whenever we find a new set of model changes where π^* is optimal and is larger than $\mathcal{E}^{MME}$, we update $\mathcal{E}^{MME}$ with $\mathcal{E}$. The resulting MME is all the possible model changes that did not appear in $\mathcal{E}^{MME}$.

Figure 6 contrasts MCE search with MME search. MCE search starts from $\mathcal{M}_h^R$, computes updates $\widehat{\mathcal{M}}$ towards $\mathcal{M}^R$ and returns the first node (indicated

in orange) where $C(\pi^*, \widehat{\mathcal{M}}) = C^*_{\widehat{\mathcal{M}}}$. MME search starts from $\mathcal{M}^R$ and moves towards $\mathcal{M}^R_h$. It finds the longest path (indicated in blue) where $C(\pi^*, \widehat{\mathcal{M}}) = C^*_{\widehat{\mathcal{M}}}$ for all $\widehat{\mathcal{M}}$ in the path. The MME (green) is the rest of the path towards $\mathcal{M}^R_h$.

2.3.3. Approximate MCE-search

Both MCEs and MMEs may be hard to compute - in the worst case it involves a search over the entire space of model differences. Thus the biggest bottleneck here is the check for optimality of a plan given a new model. A check for necessary or sufficient conditions for optimality, without actually computing optimal plans can be used as a powerful tool to further prune the search tree.

In the following section, we investigate an approximation to an MCE by employing a few simple proxies to the optimality test. By doing this we lose the completeness guarantee but improve computability. Specifically, we replace the equality test in line 12 of Algorithm 1 by the following rules –

1. $\delta_{\widehat{\mathcal{M}}}(\widehat{\mathcal{I}}, \pi^*_R) \models \widehat{\mathcal{G}}$; **and**
2. $C(\pi^*_R, \widehat{\mathcal{M}}) < C(\pi^*_R, \mathcal{M}^R_h)$ **or** $\delta_{\widehat{\mathcal{M}}}(\widehat{\mathcal{I}}, \pi^*_H) \not\models \widehat{\mathcal{G}}$; **and**
3. Each action contributes at least one causal link to π^*_R.

(1) ensures that the plan π^*_R originally computed is actually valid in the new model. (2) requires that this plan has either become better in the new model or at least that the human's expected plan π^*_H has been disproved. Finally, in (3), we ensure that for each action $a_i \in \pi^*_R$ there exists an effect p that satisfies the precondition of at least one action a_k (where $a_i \prec a_k$) and there exists no action a_j (where $a_i \prec a_j \prec a_k$) such that $p \in \mathit{eff}^-(a_j)$. Such explanations are only able to preserve local properties of a plan and hence incomplete.

Proposition 6 – Criterion (3) is a necessary condition for optimality of π^* in $\widehat{\mathcal{M}}$.

Proof. Assume that for an optimal plan π^*_R, there exists an action a_i where criterion (3) is not met. Now we can rewrite π^*_R as

$$\pi'_R = \langle a_0, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n, a_{n+1} \rangle$$

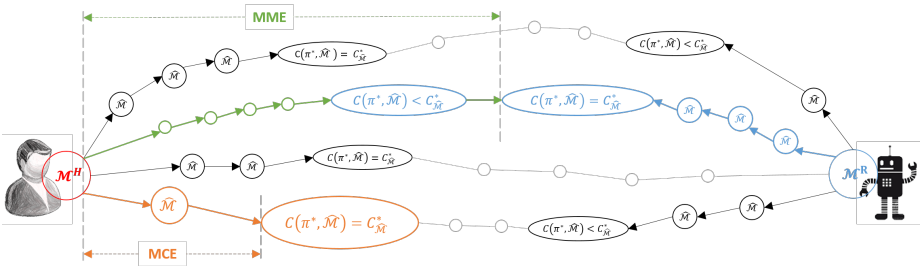


Figure 6: Illustration contrasting MCE search with MME search.

where $pre(a_0) = \emptyset$ and $\mathit{eff}^+(a_0) = \{\mathcal{I}\}$ and $pre(a_{n+1}) = \{\mathcal{G}\}$ and $\mathit{eff}(a_{n+1}) = \emptyset$. It is easy to see that $\delta_{\widehat{\mathcal{M}}}(\emptyset, \pi'_R) \models \mathcal{G}$. Now let us consider a cheaper plan $\hat{\pi'_R} = \langle a_0, a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, a_{n+1} \rangle$. Since a_i does not contribute any causal links to the original plan π^*_R, we will also have $\delta_{\widehat{\mathcal{M}}}(\emptyset, \hat{\pi}'_R) \models \mathcal{G}$. This contradicts our original assumption of π^*_R being optimal, hence proved.

3. Model Reconciliation Expansion Pack

So far we have presented the very basics of the model reconciliation framework and how it can model the explanation process in planning problems. In the process of doing so, we made several assumptions, the primary among them being that the mental model $\mathcal{M}^R_h$ is known precisely. In the following discussion, we will relax this assumption in particular, and point to other extensions by which we have since expanded on this framework.

3.1. What if the mental model is not known with certainty?

As we mentioned before, we have assumed thus far that $\mathcal{M}^R_h$ is known as a first step towards formalizing the model reconciliation process. This is hard to achieve in practice. Instead, the agent may end up having to explain its decisions with respect to a *set of possible models* which is its estimation of the human's knowledge state learned in the process of interactions. For example, consider the work in [36] where model drift is tracked via filters in the form of a set of models, or in [37] where a set of models is computed to fit to observed plan traces. Such

Algorithm 1 Search for Minimally Complete Explanations

procedure MCE-SEARCH

Input: MRP $\langle \pi^*, \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle \rangle$

Output: Explanation $\mathcal{E}^{MCE}$

Procedure:

    fringe ← `Priority_Queue()`
    c_list ← {} ▷ Closed list
    $\pi_R^* \leftarrow \pi^*$ ▷ Optimal plan being explained
    $\pi_H \leftarrow \pi$ such that $C(\pi, \mathcal{M}_h^R) = C^*_{\mathcal{M}_h^R}$ ▷ Plan expected by human
    fringe.push($\langle \mathcal{M}_h^R, \{\} \rangle$, priority = 0)
    while True **do**
        $\langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c \leftarrow$ fringe.pop($\widehat{\mathcal{M}}$)
        if $C(\pi_R^*, \widehat{\mathcal{M}}) = C^*_{\widehat{\mathcal{M}}}$ **then** return $\mathcal{E}$ ▷ Return $\mathcal{E}$ if π_R^* optimal in $\widehat{\mathcal{M}}$
        else
            c_list ← c_list ∪ $\widehat{\mathcal{M}}$
            for $f \in \Gamma(\widehat{\mathcal{M}}) \setminus \Gamma(\mathcal{M}^R)$ **do** ▷ Models that satisfy Condition 1
                $\lambda \leftarrow \langle 1, \{\widehat{\mathcal{M}}\}, \{\}, \{f\} \rangle$ ▷ Removes f from $\widehat{\mathcal{M}}$
                if $\delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\Gamma(\widehat{\mathcal{M}}), \lambda) \notin$ c_list **then**
                    fringe.push($\langle \delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\Gamma(\widehat{\mathcal{M}}), \lambda),\ \mathcal{E} \cup \lambda \rangle,\ c+1$)
            for $f \in \Gamma(\mathcal{M}^R) \setminus \Gamma(\widehat{\mathcal{M}})$ **do** ▷ Models that satisfy Condition 2
                $\lambda \leftarrow \langle 1, \{\widehat{\mathcal{M}}\}, \{f\}, \{\} \rangle$ ▷ Adds f to $\widehat{\mathcal{M}}$
                if $\delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\Gamma(\widehat{\mathcal{M}}), \lambda) \notin$ c_list **then**
                    fringe.push($\langle \delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\Gamma(\widehat{\mathcal{M}}), \lambda),\ \mathcal{E} \cup \lambda \rangle,\ c+1$)

procedure PRIORITY_QUEUE.POP($\hat{\mathcal{M}}$)

    candidates ← $\{\langle \langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c^* \rangle \mid c^* = \arg\min_c \langle \langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c \rangle\}$
    pruned_list ← {}
    $\pi_H \leftarrow \pi$ such that $C(\pi, \hat{\mathcal{M}}) = C^*_{\hat{\mathcal{M}}}$
    for $\langle \langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c \rangle \in$ candidates **do**
        if $\exists a \in \pi_R^* \cup \pi_H$ such that $\tau^{-1}(\Gamma(\widehat{\mathcal{M}}) \,\Delta\, \Gamma(\hat{\mathcal{M}})) \in \{c_a\} \cup pre(a) \cup \mathit{eff}^+(a) \cup \mathit{eff}^-(a)$ **then**
            ▷ Candidates relevant to π_R^* or π_H
            pruned_list ← pruned_list ∪ $\langle \langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c \rangle$
    if pruned_list = ∅ **then** $\langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c \sim Unif$(candidate_list)
    else $\langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c \sim Unif$(pruned_list)

Algorithm 2 Search for Minimally Monotonic Explanations

procedure MME-SEARCH

Input: MRP $\langle \pi^*, \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle \rangle$

Output: Explanation $\mathcal{E}^{MME}$

Procedure:

$\mathcal{E}^{MME} \leftarrow \{\}$

fringe $\leftarrow$ `Priority_Queue()`

c_list $\leftarrow \{\}$ ▷ Closed list

h_list $\leftarrow \{\}$ ▷ List of incorrect model changes

fringe.push($\langle \mathcal{M}^R, \{\} \rangle$, priority $= 0$)

while fringe is not empty **do**

  $\langle \widehat{\mathcal{M}}, \mathcal{E} \rangle, c \leftarrow$ fringe.pop($\widehat{\mathcal{M}}$)

  if $C(\pi^*, \widehat{\mathcal{M}}) > C^*_{\widehat{\mathcal{M}}}$ **then**

    h_list $\leftarrow$ h_list $\cup$ $(\Gamma(\widehat{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^R))$ ▷ Updating h_list

  else

    c_list $\leftarrow$ c_list $\cup$ $\widehat{\mathcal{M}}$

    for $f \in \Gamma(\widehat{\mathcal{M}}) \setminus \Gamma(\mathcal{M}_h^R)$ **do** ▷ Models that satisfy Condition 1

      $\lambda \leftarrow \langle 1, \{\widehat{\mathcal{M}}\}, \{\}, \{f\} \rangle$ ▷ Removes f from $\widehat{\mathcal{M}}$

      if $\delta_{\mathcal{M}^R, \mathcal{M}_h^R}(\Gamma(\widehat{\mathcal{M}}), \lambda) \notin$ c_list
        and $\nexists S$ s.t. $(\Gamma(\widehat{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^R)) \supseteq S \in$ h_list **then** ▷ Proposition 3

        fringe.push($\langle \delta_{\mathcal{M}^R, \mathcal{M}_h^R}(\Gamma(\widehat{\mathcal{M}}), \lambda),\ \mathcal{E} \cup \lambda \rangle,\ c+1$)

        $\mathcal{E}^{MME} \leftarrow \max_{|\cdot|}\{\mathcal{E}^{MME}, \mathcal{E}\}$

    for $f \in \Gamma(\mathcal{M}_h^R) \setminus \Gamma(\widehat{\mathcal{M}})$ **do** ▷ Models that satisfy Condition 2

      $\lambda \leftarrow \langle 1, \{\widehat{\mathcal{M}}\}, \{f\}, \{\} \rangle$ ▷ Adds f from $\widehat{\mathcal{M}}$

      if $\delta_{\mathcal{M}^R, \mathcal{M}_h^R}(\Gamma(\widehat{\mathcal{M}}), \lambda) \notin$ c_list
        and $\nexists S$ s.t. $(\Gamma(\widehat{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^R)) \supseteq S \in$ h_list **then** ▷ Proposition 3

        fringe.push($\langle \delta_{\mathcal{M}^R, \mathcal{M}_h^R}(\Gamma(\widehat{\mathcal{M}}), \lambda),\ \mathcal{E} \cup \lambda \rangle,\ c+1$)

        $\mathcal{E}^{MME} \leftarrow \max_{|\cdot|}\{\mathcal{E}^{MME}, \mathcal{E}\}$

$\mathcal{E}^{MME} \leftarrow (\Gamma(\widehat{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^R)) \setminus \mathcal{E}^{MME}$

return $\mathcal{E}^{MME}$

uncertainty or incompleteness over a model can be represented in the form of *annotated* models or `APDDL`. [37] In addition to the standard preconditions and effects associated with actions, it introduces the notion of *possible* preconditions and effects which may or may not be realized in practice.

Definition: **An Annotated Model** is the tuple $\mathbb{M} = \langle \mathbb{D}, \mathbb{I}, \mathbb{G} \rangle$ with a domain $\mathbb{D} = \langle F,\ \mathbb{A} \rangle$ – where F is a finite set of fluents that define a state $s \subseteq F$, and $\mathbb{A}$ is a finite set of annotated actions – and annotated initial and goal states $\mathbb{I} = \langle \mathcal{I}^0, \mathcal{I}^+ \rangle,\ \mathbb{G} = \langle \mathcal{G}^0, \mathcal{G}^+ \rangle;\ \mathcal{I}^0, \mathcal{G}^0, \mathcal{I}^+, \mathcal{G}^+ \subseteq F$. Action $a \in \mathbb{A}$ is a tuple $\langle c_a, \mathit{pre}(a), \widetilde{\mathit{pre}}(a), \mathit{eff}^{\pm}(a), \widetilde{\mathit{eff}}^{\pm}(a) \rangle$ where c_a is the cost and, in addition to its *known* preconditions and add/delete effects $\mathit{pre}(a), \mathit{eff}^{\pm}(a), \subseteq F$ each action also contains *possible preconditions* $\widetilde{\mathit{pre}}(a) \subseteq F$ containing propositions that it *might* need as preconditions, and *possible add (delete) effects* $\widetilde{\mathit{eff}}^{\pm}(a) \subseteq F$) containing propositions that it *might* add (delete, respectively) after execution. $\mathcal{I}^0, \mathcal{G}^0$ (and $\mathcal{I}^+, \mathcal{G}^+$) are the known (and possible) parts of the initial and goal states.

Each possible condition $f \in \widetilde{\mathit{pre}}(a) \cup \widetilde{\mathit{eff}}^{\pm}(a)$ has an associated probability $p(f)$ denoting how likely it is to be a known condition in the ground truth model – i.e. $p(f)$ measures the confidence with which that condition has been learned. The sets of known and possible conditions of a model $\mathcal{M}$ are denoted by $\mathbb{S}_k(\mathcal{M})$ and $\mathbb{S}_p(\mathcal{M})$ respectively. An *instantiation* of an annotated model $\mathbb{M}$ is a classical planning model where a subset of the possible conditions have been realized, and is thus given by the tuple $\mathit{inst}(\mathbb{M}) = \langle \mathcal{D}, \mathcal{I}, \mathcal{G} \rangle$ with domain $\mathcal{D} = \langle F, A \rangle$, initial and goal states $\mathcal{I} = \mathcal{I}^0 \cup \chi;\ \chi \subseteq \mathcal{I}^+$ and $\mathcal{G} = \mathcal{G}^0 \cup \chi;\ \chi \subseteq \mathcal{G}^+$ respectively, and action $A \ni a = \langle c_a, \mathit{pre}(a) \leftarrow \mathit{pre}(a) \cup \chi;\ \chi \subseteq \widetilde{\mathit{pre}}(a), \mathit{eff}^{\pm}(a) \leftarrow \mathit{eff}^{\pm}(a) \cup \chi;\ \chi \subseteq \widetilde{\mathit{eff}}^{\pm}(a) \rangle$. Clearly, given an annotated model with k possible conditions, there may be 2^k such instantiations, which forms its *completion set.* [37]

Definition: **Likelihood** $\mathcal{L}$ of instantiation $\mathit{inst}(\mathbb{M})$ of an annotated model $\mathbb{M}$ is:

$$\mathcal{L}(\mathit{inst}(\mathbb{M})) = \prod_{f \in \mathbb{S}_p(\mathbb{M}) \wedge \mathbb{S}_k(\mathit{inst}(\mathbb{M}))} p(f) \quad \times \quad \prod_{f \in \mathbb{S}_p(\mathbb{M}) \setminus \mathbb{S}_k(\mathit{inst}(\mathbb{M}))} (1 - p(f))$$

As discussed before, such models turn out to be especially useful for the

representation of human (mental) models learned from observations, where uncertainty after the learning process can be represented in terms of model annotations. [37, 36] Let $\mathbb{M}_H^R$ be the culmination of a model learning process and $\{\mathcal{M}_{h_i}^R\}_i$ be the completion set of $\mathbb{M}_H^R$. One of these models is the actual ground truth (i.e. the human's real mental model). We refer to this as $g(\mathbb{M}_H^R)$. We will explore now how this representation will allow us to compute *conformant explanations* that can explain with respect to all possible mental models and *conditional explanations* that engage the explainee in dialogue to minimize the size of the completion set to compute shorter explanations.[9]

3.1.1. Conformant Explanations

In this situation, the robot can try to compute MCEs for each possible configuration. However, this can result in situations where the explanations computed for individual models independently are not consistent across all possible target domains. Thus, in the case of model uncertainty, such an approach cannot guarantee that the resulting explanation will be acceptable.

Instead, we want to find an explanation such that $\forall i\ \pi^*_{\widehat{\mathcal{M}}_{h_i}^R} \equiv \pi^*_{\mathcal{M}^R}$ (as shown in Figure 7). This is a single model update that makes the given plan optimal (and hence explained) in all the updated domains (or in all possible domains). At first glance, it appears that such an approach, even though desirable, might turn out to be prohibitively expensive especially since solving for a *single* MCE involves search in the model space where each search node is an optimal planning problem. However, it turns out that the same search strategy can be employed here as well by representing the human mental model as an *annotated* model. Condition (3) for an MCE (c.f. Section 2) now becomes –

[9]The purpose of this section is to demonstrate how existing notions of conditional and conformant solutions in planning can be adopted for the explanation process equally well in the presence of uncertainty over the human mental model. While there are significant differences between how conditional or conformant explanations work with respect to their planning counterparts, it may be worth exploring the state-of-the-art [38, 39] in those fields to further develop on the concepts introduced in the paper.

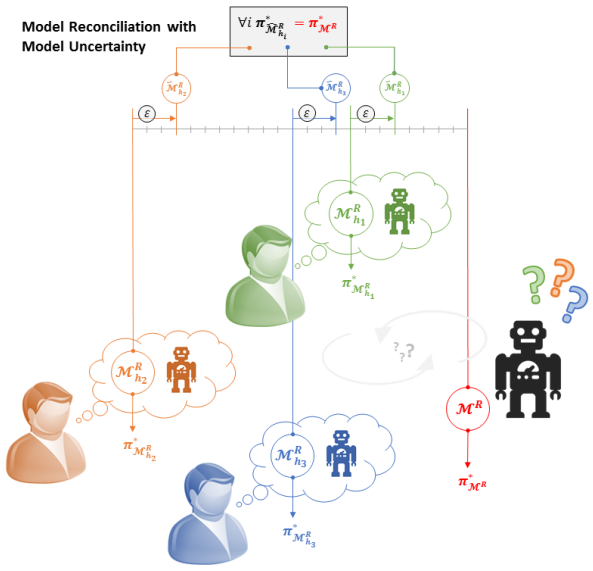


Figure 7: Model reconciliation in the presence of model uncertainty or multiple explainees.

(3) $C(\pi, g(\mathbb{M}_h^R)) = C^*_{g(\mathbb{M}_h^R)}$

This is hard to achieve since it is not known which is the actual mental model of the human. So we want to preserve the optimality criterion for all (or as many) instantiations of the incomplete estimation of the mental model. Keeping this in mind, we define *robustness* of an explanation for an incomplete mental models as the probability mass of models where it is a valid explanation.

Definition: **Robustness** of an explanation $\mathcal{E}$ is given by –

$$R(\mathcal{E}) = \sum_{inst(\widehat{\mathcal{M}}_h^R) \text{ s.t. } C(\pi, inst(\widehat{\mathcal{M}}_h^R)) = C^*_{inst(\widehat{\mathcal{M}}_h^R)}} \mathcal{L}(inst(\widehat{\mathcal{M}}_h^R))$$

Definition: **A Conformant Explanation** is such that $R(\mathcal{E}) = 1$.

A conformant explanation thus ensures that the given plan is explained in all the models in the completion set of the human model. Let's look at an example. Consider again the USAR domain (Figure 8), the robot is now at P1 (blue) and needs to collect data from P5. While the commander understands the goal, she

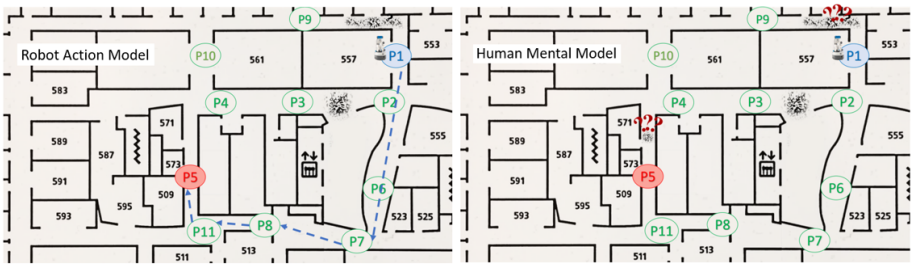


Figure 8: Back to our USAR scenario: the robot plan is marked in blue and uncertain parts of the human model is marked with red question marks.

is under the false impression that the paths from P1 to P9 and P4 to P5 are unusable (red question marks). She is also unaware of the robot's inability to use its hands. On the other hand, while the robot does not have a complete picture of her mental model, it understands that any differences between the models are related to (1) the path from P1 to P9; (2) the path from P4 to P5; (3) its ability to use its hands; and (4) whether the it needs its arm to clear rubble. Thus, from the robot's perspective, the mental model can be one of sixteen possible models (one of which is the actual one). Here, a conformant explanation for the optimal robot plan (blue) is as follows –

```
Explanation >> remove-known-INIT-has-add-effect-hand_capable
Explanation >> add-annot-clear_passage-has-precondition-hand_capable
Explanation >> remove-annot-INIT-has-add-effect-clear_path P1 P9
```

3.1.2. Model-Space Search for Conformant Explanations

As we discussed before, we cannot launch an MCE-search for each possible mental model separately, both for issues of complexity and consistency of the solutions. However, in the following discussion, we will show how we can reuse the model space search from the previous section with a compilation trick.

We begin by defining two models – the most relaxed model possible $\mathcal{M}_{max}$ and the least relaxed one $\mathcal{M}_{min}$. The former is the model where all the possible add effects and none of the possible preconditions and deletes hold, the state has

all the possible conditions set to true, and the goal is the smallest one possible; while in the latter all the possible preconditions and deletes and none of the possible adds are realized and with the minimal start state and the maximal goal. This means that, if a plan is executable in $\mathcal{M}_{min}$ it will be executable in all the possible models. Also, if this plan is optimal in $\mathcal{M}_{max}$, then it must be optimal throughout the set. Of course, such a plan may not exist, but we are not trying to find one either. Instead, we are trying to find a set of model updates which when applied to the annotated model, produce a new set of models where a *given* plan is optimal. In providing these model updates, we are in effect reducing the set of possible models to a smaller set. The new set need not be a subset of the original set of models but will be equal or smaller in size to the original set. For any given annotated model, such an explanation always exists (entire model difference in the worst case), and we intend to find the smallest one. $\mathbb{M}_h^R$ thus affords the following two models –

$\mathcal{M}_{max} = \langle \mathcal{D}, \mathcal{I}, \mathcal{G} \rangle$ with domain $\mathcal{D} = \langle F, A \rangle$ and

- initial state $\mathcal{I} \leftarrow \mathcal{I}^0 \cup \mathcal{I}^+$; given $\mathbb{I}$
- goal state $\mathcal{G} \leftarrow \mathcal{G}^0$; given $\mathbb{G}$
- $\forall a \in A$
    - $pre(a) \leftarrow pre(a);\ a \in \mathbb{A}$
    - $\mathit{eff}^+(a) \leftarrow \mathit{eff}^+(a) \cup \widetilde{\mathit{eff}}^+(a);\ a \in \mathbb{A}$
    - $\mathit{eff}^-(a) \leftarrow \mathit{eff}^-(a);\ a \in \mathbb{A}$

$\mathcal{M}_{min} = \langle \mathcal{D}, \mathcal{I}, \mathcal{G} \rangle$ with domain $\mathcal{D} = \langle F, A \rangle$ and

- initial state $\mathcal{I} \leftarrow \mathcal{I}^0$; given $\mathbb{I}$
- goal state $\mathcal{G} \leftarrow \mathcal{G}^0 \cup \mathcal{G}^+$; given $\mathbb{G}$
- $\forall a \in A$
    - $pre(a) \leftarrow pre(a) \cup \widetilde{pre}(a);\ a \in \mathbb{A}$
    - $\mathit{eff}^+(a) \leftarrow \mathit{eff}^+(a);\ a \in \mathbb{A}$
    - $\mathit{eff}^-(a) \leftarrow \mathit{eff}^-(a) \cup \widetilde{\mathit{eff}}^-(a);\ a \in \mathbb{A}$

As explained before, $\mathcal{M}_{max}$ is a model where all the add effects hold and it is easiest to achieve the goal, and similarly $\mathcal{M}_{min}$ is the model where it is the hardest to achieve the goal. These definitions might end up creating inconsistencies (e.g. in an annotated `BlocksWorld` domain, the `unstack` action may have add effects to make the block both `holding` and `ontable` at the same time), but the model reconciliation process will take care of these.

Proposition 7 – For a given MRP $\Psi = \langle \pi, \langle \mathcal{M}^R, \mathbb{M}_h^R \rangle \rangle$, if the plan π is optimal in $\mathcal{M}_{max}$ and executable in $\mathcal{M}_{min}$,then conditions (1) and (3) from the definition of valid model reconciliation explanation (as defined in Section 2) hold for all i.

This now becomes the new criterion to satisfy in the course of search for an MCE for a set of models. We again reuse the state representation in Section 2.1. We start the `MEGA*-Conformant` search (Algorithm 3) by first creating the corresponding $\mathcal{M}_{max}$ and $\mathcal{M}_{min}$ model for the given annotated model $\mathbb{M}_H^R$. While the goal test for the original MCE only included an optimality test, here we need to both check the optimality of the plan in $\mathcal{M}_{max}$ and verify the correctness of the plan in $\mathcal{M}_{min}$. As stated in Proposition 7, the plan is only optimal in the entire set of possible models if it satisfies both tests. Since the correctness of a given plan can be verified in polynomial time with respect to the plan size, this is a relatively easy test to perform.

The other important point of difference between the algorithm mentioned above and the original MCE is how we calculate the applicable model updates. Here we consider the superset of model differences between the robot model and $\mathcal{M}_{min}$ and the differences between the robot model and $\mathcal{M}_{max}$. This could potentially mean that the search might end up applying a model update that is already satisfied in one of the models but not in the other. Since all the model update actions are formulated as set operations, the original MRP formulation can handle this without any further changes. The models obtained by applying the model update to $\mathcal{M}_{min}$ and $\mathcal{M}_{max}$ are then pushed to the open queue.

Proposition 8 – $\mathcal{M}_{max}$ and $\mathcal{M}_{min}$ only need to be computed once – i.e. with a

Algorithm 3 `MEGA*-Conformant`

procedure MCE-SEARCH
Input: MRP $\langle \pi^*, \langle \mathcal{M}^R, \mathbb{M}_h^R \rangle \rangle$
Output: Explanation $\mathcal{E}^{MCE}$
Procedure:
    fringe ← `Priority_Queue()`
    c_list ← {} ▷ Closed list
    $\pi_R^* \leftarrow \pi^*$ ▷ Optimal plan being explained
    $\mathcal{M}_{max},\ \mathcal{M}_{min} \leftarrow (\mathbb{M}_h^R)$ ▷ Proposition 8
    fringe.push($\langle \mathcal{M}_{min}, \mathcal{M}_{max}, \{\} \rangle,\ \text{priority} = 0$)
    while True **do**
        $\langle \widehat{\mathcal{M}}_{min}, \widehat{\mathcal{M}}_{max}, \mathcal{E} \rangle, c \leftarrow$ fringe.pop()
        if $C(\pi_R^*, \widehat{\mathcal{M}}_{max}) = C^*_{\widehat{\mathcal{M}}_{max}} \wedge \delta(\mathcal{I}_{\widehat{\mathcal{M}}_{min}}, \pi_R^*) \models \mathcal{G}_{\widehat{\mathcal{M}}_{min}}$ **then**
            return $\mathcal{E}$ ▷ Proposition 7
        else
            c_list ← c_list $\cup \langle \widehat{\mathcal{M}}_{max}, \widehat{\mathcal{M}}_{min} \rangle$
            for $f \in \{\Gamma(\widehat{\mathcal{M}}_{min}) \cup \Gamma(\widehat{\mathcal{M}}_{max})\} \setminus \Gamma(\mathcal{M}^R)$ **do**
                $\lambda \leftarrow \langle 1, \langle \widehat{\mathcal{M}}_{min}, \widehat{\mathcal{M}}_{max} \rangle, \{\}, \{f\} \rangle$ ▷ Removes f from $\widehat{\mathcal{M}}$
                if $\delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max}) \rangle, \lambda) \notin$ c_list **then**
                    fringe.push($\langle \delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max}) \rangle, \lambda), \mathcal{E} \cup \lambda \rangle,\ c + 1$)
            for $f \in \Gamma(\mathcal{M}^R) \setminus \{\Gamma(\widehat{\mathcal{M}}_{min}) \cup \Gamma(\widehat{\mathcal{M}}_{max})\}$ **do**
                $\lambda \leftarrow \langle 1, \{\langle \widehat{\mathcal{M}}_{min}, \widehat{\mathcal{M}}_{max} \rangle\}, \{f\}, \{\} \rangle$ ▷ Adds f to $\widehat{\mathcal{M}}$
                if $\delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max}) \rangle, \lambda) \notin$ c_list **then**
                    fringe.push($\langle \delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max}) \rangle, \lambda), \mathcal{E} \cup \lambda \rangle,\ c + C_\lambda$)

model update $\mathcal{E}$ to $\mathbb{M}$: $\mathcal{M}_{max} \leftarrow \mathcal{M}_{max} + \mathcal{E}$ and $\mathcal{M}_{min} \leftarrow \mathcal{M}_{min} + \mathcal{E}$.

These models form the new $\mathcal{M}_{min}$ and $\mathcal{M}_{max}$ models for the set of models obtained by applying the current set of model updates to the original annotated model. This proposition ensures that we no longer have to keep track of the current list of models or recalculate $\mathcal{M}_{min}$ and $\mathcal{M}_{max}$ for the new set.

3.1.3. Contingent Explanations

Conformant explanations can contain superfluous information – i.e. asking the human to remove non-existent conditions or add existing ones. In the previous example, the second explanation (regarding the need of the hand to clear rubble) was already known to the human and was thus superfluous information. Such redundant information can be annoying and may end up reducing the human's trust in the robot. This can be avoided by –

- Increasing the cost of model updates involving uncertain conditions relative to those involving known preconditions or effects. This ensures that the search prefers explanations that contain known conditions. By definition, such explanations will not have superfluous information.

- However, sometimes such explanations may not exist. Instead, we can convert conformant explanations into *conditional* ones. This can be achieved by turning each model update for an annotated condition into a question and only provide an explanation if the human's response warrants it – e.g. instead of asking the human to update the precondition of `clear_passage`, the robot can first ask if the human thinks that action has a precondition `hand_usable`. Thus, one way of removing superfluous explanations is to reduce the size of the completion set by gathering information from the human. Consider the following exchange in the USAR scenario –

```
R : Are you aware that the path from P1 to P4 has collapsed?
H : Yes.
> R realizes the plan is optimal in all possible models.
> It does not need to explain further.
```

If the robot knew that the human thought that the path from P1 to P4 was collapsed, it would know that the robot's plan is already optimal in the human mental model and hence be required to provide no further explanation. This form of explanations can thus clearly be used to cut down on the size of conformant explanations by reducing the size of the completion set.

Definition: **A Conditional Explanation** is represented by a policy that maps the annotated model (represented by a $\mathcal{M}_{min}$ and $\mathcal{M}_{max}$ model pair) to either a question regarding the existence of a condition in the human ground model or a model update request. The resultant annotated model is produced, by either applying the model update directly into the current model or by updating the model to conform to human's answer regarding the existence of the condition.

In asking questions such as these, the robot is trying to exploit the human's (lack of) knowledge of the problem in order to provide more concise explanations. This can be construed as a case of lying by omission and can raise interesting ethical considerations [29]. Humans, during an explanation process, tend to undergo this same "selection" process [13] as well in determining which of the many reasons that could explain an event is worth highlighting.

Modified AO-search to find Conditional Explanations.* We can generate conditional explanations by either performing post-processing on conformant explanations or by performing an AND-OR graph search with *AO**[40]. *AO** is a heuristic search procedure for acyclic AND-OR graph search, that is guaranteed to identify the optimal solution, provided the heuristics are admissible and monotonic.

Here each model update related to a known condition forms an OR successor node while each *possible* condition can be applied on the current state to produce a pair of AND successors, where the first node reflects a node where the annotated condition holds while the second one represents the state where it does not. So the number of possible conditions reduces by one in each one of these AND successor nodes. This AND successor relates to the answers the human could

potentially provide when asked about the existence of that particular possible condition. Note that this AND-OR graph will not contain any cycles as we only provide model updates that are consistent with the robot model and hence we can directly use the AO^* search here.

Throughout this section, we will use h=0 as our heuristic and unit cost for explanations and queries. AO^* search can be understood to be operating in two distinct phases, in the first phase the algorithm considers the current best partial solution and identifies a node to expand. This is a top-down operation and the next phase is a bottom-up cost and label revision stage. Here both the label and the cost of the newly expanded node are propagated back up the graph. The acyclicity of our setting ensures that this backward propagation can be easily carried out. This is done by adding any parents of an updated node into the set S and the procedure continues until the set is empty. All goal nodes (i.e. explanations holds in all remaining models) are marked with SUCCESS label. Each parent node receives the label of the successor node with the minimum value. In the case of an AND successor, the parent only receives the SUCCESS label if all the possible children has the SUCCESS label. The search ends when the root node is assigned the SUCCESS label.

Unfortunately, if we used the standard AO^* search, it will not produce a conditional explanation that contains this "less robust" explanation as one of the potential branches in the conditional explanation. This is because, in the above example, if the human had said that the path was free, the robot would need to revert to the original conformant explanation. Thus the cost of the subtree containing this solution will be higher than the one that only includes the original conformant explanation.

To overcome this shortcoming, we introduce a discounted version of the AO^* search where the cost contributed by a pair of AND successors is calculated as –

$$min(node1.h\_val,\ node2.h\_val) + \gamma * max(node1.h\_val,\ node2.h\_val)$$

where node1 and node2 are the successor nodes and node1.h_val, node2.h_val are their respective h-values. Here γ represents the discount fact and controls

how much the search values short paths in its solution subtree. When $\gamma = 1$, the search becomes standard AO^* search and when $\gamma = 0$, the search myopically optimizes for short branches (at the cost of the depth of the solution subtree). The rest of the algorithm stays the same as the standard AO^* search. Though with this modification, AO^* is no longer guarateed to generate optimal solutions when $\gamma \neq 1$. The pseudocode is provided in Algorithm 4.

3.1.4. Anytime Explanations

As we will see later in the evaluations, both the algorithms discussed above can be computationally expensive, in spite of the compilation trick to reduce the set of possible models to two representative models. However, as we did previously with MCEs, we can also shoot for an approximate solution by relaxing the minimality requirement of explanation to achieve much shorter explanation generation time when required. For this we introduce an anytime depth first explanation generation algorithm. Here, for each state, the successor states include all the nodes that can be generated by applying the model edit actions on all the known predicates and two possible successors for each possible condition – one where the condition holds and one where it does not. Once the search reaches a goal state (a new model where the target plan is optimal throughout its completion set), it queries the human to see if the assumptions it has made regarding possible conditions hold in the human mental model (the list of model updates made related to possible conditions). If all the assumptions hold in the human model, then we return the current solution as the final explanation (or use the answers to look for smaller explanations), else continue the search after pruning the search space using the answers provided by the human. Such approaches may also be able to facilitate iterative presentation of model reconciliation explanations to the human. [41]

The pruning can be performed efficiently by keeping track of all the human answers and enforcing these specifications only at the time of expansion of new nodes. Algorithm 5 presents a depth-first search approach for an anytime solution. Here we add an additional variable $\mathcal{A}$ to the search node to keep

Algorithm 4 MEGA*-Conditional

procedure AO*-SEARCH
Input: MRP $\langle \pi^*, \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle \rangle, \gamma$
Output: Explanation $\mathcal{E}^{MCE}$
Procedure:
    c_list ← {} ▷ Closed list
    π_R^* ← π^* ▷ Optimal plan being explained
    M_{min}^H ←MIN_MODEL(M^H) ▷ Generates M_{min} as per definition
    M_{max}^H ←MAX_MODEL(M^H) ▷ Generates M_{max} as per definition
    G← Node($\langle \mathcal{M}_{min}^H, \mathcal{M}_{max}^H, \{\} \rangle$)
    while G.label != Success **do**
        current_node ← Unexpanded_node(G) ▷ Return an unexpanded node in the current best path
        S ← {curent_node}
        curent_node.successors ← GetSuccessors(current_node)
        while S not empty() **do**
            node ← S.pop() ▷ Refer to [40] to see how to prioritize which nodes to remove
            min_val ← 0
            label ← None
            for succ in node.successors **do**
                if succ is a OR_Succ **then**
                    node1 ← succ
                    if min_val > node1.h_val **then**
                        min_val = node1.h_val
                        label = node1.label
                if succ is a AND_Succ **then**
                    node1, node2 ← succ
                    tmp_val = min(node1.h_val, node2.h_val) + γ * max(node1.h_val, node2.h_val)
                    if min_val > tmp_val **then**
                        min_val = tmp_val
                        if node1.label == node2.label **then**
                            label = node1.label
                node.label = label
                nodel.h_val = 1 + min_val
            Add all parents of node to S

procedure GETSUCCESSORS(node, $\mathcal{M}^R$)
    min_state, max_state ← node.state
    Known_predicates ← $\Gamma(min\_state) \cap \Gamma(max\_state)$
    Possible_predicates ← $\Gamma(min\_state) \Delta \Gamma(max\_state)$
    OR_actions_deletes ← {Known_predicates \$\Gamma(\mathcal{M}^R)$}
    OR_actions_adds{$\Gamma(\mathcal{M}^R)$\Known_predicates}
    AND_actions ← Possible_predicates
    Succ_nodes ← Set()
    for a ∈ OR_actions_adds **do**
        tmp_node = Node($\langle \Gamma^{-1}(min\_state \cup a), \Gamma^{-1}(max\_state \cup a) \rangle$)
        tmp_node ← Evaluate_Node(tmp_node)
        Succ_nodes.push(OR_succ(tmp_node)))
    for a ∈ OR_actions_deletes **do**
        tmp_node = Node($\langle \Gamma^{-1}(min\_state \setminus a), \Gamma^{-1}(max\_state \setminus a) \rangle$)
        tmp_node ← Evaluate_Node(node)
        Succ_nodes.push(OR_succ(tmp_node))
    for a ∈ AND_actions **do**
        tmp_node1 = Node($\langle \Gamma^{-1}(min\_state \cup a), \Gamma^{-1}(max\_state \cup a) \rangle$)
        tmp_node2 = Node($\langle \Gamma^{-1}(min\_state \setminus a), \Gamma^{-1}(max\_state \setminus a) \rangle$)
        tmp_node1 ← Evaluate_Node(tmp_node1)
        tmp_node2 ← Evaluate_Node(tmp_node2)
        Succ_nodes.push(AND_succ(tmp_node1, tmp_node2))
    return Succ_nodes

procedure EVALUATE_NODE(node)
    if Check_For_Goal(node) **then** ▷ Refer to MEGA*-Conformant for goal test
        node.h_val ← 0
        node.label ← SUCCESS
    else
        node.h_val ← heuristic(node)
    return node

Algorithm 5 `MEGA*-Anytime`

procedure ANYTIME-EXPLANATION
Input: MRP $\langle \pi^*, \langle \mathcal{M}^R, \mathbb{M}_h^R \rangle \rangle$
Output: Explanation $\mathcal{E}$
Procedure:
    fringe $\leftarrow$ `Stack()`
    $\pi_R^* \leftarrow \pi^*$ ▷ Optimal plan being explained
    $\mathcal{M}_{max},\ \mathcal{M}_{min}\ \leftarrow (\mathbb{M}_h^R)$ ▷ Proposition 8
    $\mathcal{A} \leftarrow \{\}$ ▷ Current assumptions
    fringe.push($\langle \mathcal{M}_{min}, \mathcal{M}_{max}, \mathcal{A}, \{\} \rangle$)
    while True **do**
        $\langle \widehat{\mathcal{M}}_{min}, \widehat{\mathcal{M}}_{max}, \mathcal{A}, \mathcal{E} \rangle \leftarrow$ fringe.pop()
        if $C(\pi_R^*, \widehat{\mathcal{M}}_{max}) = C^*_{\widehat{\mathcal{M}}_{max}} \wedge \delta(\mathcal{I}_{\widehat{\mathcal{M}}_{min}}, \pi_R^*) \models \mathcal{G}_{\widehat{\mathcal{M}}_{min}}$ **then**
            $\mathcal{A}_{valid}, \mathcal{A}_{invalid} \leftarrow$ TEST_ASSUMPTION($\mathcal{A}$)
            $\mathcal{A}_{valid} \leftarrow \mathcal{A} \setminus \mathcal{A}_{invalid}$
            if $|\mathcal{A}_{invalid}| = 0$ **then**
                return $\mathcal{E}$ ▷ Proposition 7
            else
                UPDATE_STACK(fringe, $\mathcal{A}_{valid}, \mathcal{A}_{invalid}$)
        else
            for $f \in \{\Gamma(\widehat{\mathcal{M}}_{min}) \cup \Gamma(\widehat{\mathcal{M}}_{max})\} \setminus \Gamma(\mathcal{M}^R)$ **do**
                $\lambda \leftarrow \langle 1, \langle \widehat{\mathcal{M}}_{min}, \widehat{\mathcal{M}}_{max} \rangle, \{\}, \{f\} \rangle$ ▷ Removes f from $\widehat{\mathcal{M}}$
                if $f \notin \{\Gamma(\widehat{\mathcal{M}}_{min}) \cap \Gamma(\widehat{\mathcal{M}}_{max})\} \setminus \Gamma(\mathcal{M}^R)$ **then**
                    $\mathcal{A} \leftarrow \mathcal{A} \cup f$ ▷ Add to assumptions if possible condition
                fringe.push($\langle \delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max}) \rangle, \lambda), \mathcal{E} \cup \lambda \rangle, \mathcal{A}$)
            for $f \in \Gamma(\mathcal{M}^R) \setminus \{\Gamma(\widehat{\mathcal{M}}_{min}) \cup \Gamma(\widehat{\mathcal{M}}_{max})\}$ **do**
                $\lambda \leftarrow \langle 1, \{\langle \widehat{\mathcal{M}}_{min}, \widehat{\mathcal{M}}_{max} \rangle, \{f\}, \{\}\rangle$ ▷ Adds f to $\widehat{\mathcal{M}}$
                fringe.push($\langle \delta_{\mathcal{M}_h^R, \mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max}) \rangle, \lambda), \mathcal{E} \cup \lambda \rangle,\ \mathcal{A}$)

track of the possible assumptions that we have made for any given search path. The `TEST_ASSUMPTION` denotes the function responsible for testing the set of assumptions during the goal test. `TEST_ASSUMPTION` returns the set of assumptions that were invalidated by the human $\mathcal{A}_{invalid}$ and we can return the current search path as a solution if the invalid set is empty. We will use the validated and invalidated assumption to update our current search stack (via the `UPDATE_STACK` function).

3.2. What if there are multiple humans in the loop?

While generating explanations for a *set of models*, the robot is essentially trying to cater to multiple human models at the same time. We posit then that the same approaches can be adopted to situations when there are *multiple humans* in the loop instead of a single human whose model is not known with certainty. As before, computing separate explanations for each agent can result in situations where the explanations computed for individual models independently are not consistent across all the possible target domains. In the case of multiple teammates being explained to, this may cause confusion and loss of trust, especially in teaming with humans who are known [42] to rely on shared mental models. Thus *conformant explanations* can be useful in dealing with not only model uncertainty but also model multiplicity.

In order to do this, from the set of target human mental models we construct an annotated model so that *the preconditions and effects that appear in all target models become necessary ones, and those that appear in just a subset are possible ones*. As before, we find a single explanation that is a satisfactory explanation for the entire set of models, without having to repeat the standard MRP process over all possible models while coming up with an explanation that can satisfy all of them and thus establish common ground.

While the explanation generation technique may be equivalent, the *explanation process* may be different depending on the setup. For example, while in the case of model uncertainty, the safest approach might be to generate explanations that work for the largest set of possible models, in scenarios with multiple explainees, the robot may have to decide whether it needs to save computational and communication time by generating one explanation to fit all models, or if it needs to tailor the explanation to each human. This choice may depend on the particular domain and the nature of the teaming relationship with the human.

In order to understand this better with the use of an example, we go back to our USAR domain, now with *two* human teammates, one external and one internal. The robot is now positioned at `P1` and is expected to collect data from location `P5`. Before the robot can perform its `surveil` action, it needs to obtain

a set of tools from the internal human agent. The human agent is initially located at P10 and is capable of traveling to reachable locations to meet the robot for the handover. Here the external commander incorrectly believes that the path from P1 to P9 is clear, while the one from P2 to P3 is closed. The internal human agent, on the other hand, not only believes in the errors mentioned above but is also under the assumption that the path from P4 to P5 is not traversable. Due to these different initial states, each of these agents ends up generating a different optimal plan. The plan expected by the external commander requires the robot to move to location P10 (via P9) to meet the human. After collecting the package from the internal agent, the commander expects it to set off to P5 via P4. The internal agent, on the other hand, believes that he needs to travel to P9 to hand over the package. As he believes that the corridor from P4 to P5 is blocked, he expects the robot to take the longer route to P5 through P6, P7, and P8 (orange). Finally, the optimal plan for the robot (blue) involves the robot meeting the human at P4 on its way to P5. MEGA*-Conformant finds the smallest explanation which explains this plan to both humans.

In this particular case, since the models differ from each other with respect to their initial states, the initial state of the corresponding annotated model is –

$\mathcal{I}^0 =$ {(at_P1), (at_human P10), ..., (clear_path P10 P9), (clear_path P9 P1)}

$\mathcal{I}^+ =$ {(clear_path P4 P5), (collapsed_path P4 P5)}

where $\mathcal{I}^+$ represents the state fluents that may or may not hold in human's model. The corresponding initial states for M_{min} and M_{max} will be as follows –

$\mathcal{I}_{min} =$ {(at_P1), (at_human P10), ..., (clear_path P10 P9), (clear_path P9 P1)}

$\mathcal{I}_{max} = \mathcal{I}_{min} \cup \{(clear\_pathP4P5), (collapsed\_pathP4P5)\}$

MEGA*-Conformant thus generates the following explanation –

```
Explanation >> add-INIT-has-clear_path P4 P5
Explanation >> remove-INIT-has-clear_path P1 P9
Explanation >> add-INIT-has-clear_path P2 P3
```

The first update specifically helps the internal to understand that the robot can indeed reach the goal through `P4`, while the next two are relevant for both the explainees to explain why they should meet at `P4` rather instead.

3.3. What if the mental model is represented differently?

Even though we accounted for uncertainty in the mental model, most of the above discussion has focused on generating explanations in cases where both the human and the robot understands the task at the same granularity. Applying model reconciliation without acknowledging the difference in the level of "expertise" can lead to confusion and information overload. Indeed, we had previously acknowledged how doctors explain differently to their colleagues than their patients. In fact, explanation generation techniques for machine learning systems have explicitly modeled this difference. [43, 44] In recent work [45], we have expanded model reconciliation framework to generate explanations when the human has access to only an abstract version of the model of the robot. Specifically, we focused on state abstractions where the abstract model was formed by projecting out a certain subset of state fluents [46], though the principles carry over to other types of abstraction as well (e.g. temporal abstractions of the types discussed in [47]).

As we noted at the beginning, the concept of model reconciliation is not particularly confined to a particular decision making model. Other works have extended the framework beyond planning problems, such as in logic programs. [48, 49] Similar ideas of model mismatch has been explored in the summarization of policies in markov decision processes[50]. Indeed, in recent work, we have been exploring cases where the explicit representation of the mental model can be dropped [51] and a desired explanation can be learned through interactions. Such approaches, in addition to abstraction techniques [45, 52] discussed above, also implicitly account for the inferential capability of the explainee.

4. Empirical Evaluations

We performed a set of empirical evaluation to evaluate the computational characteristics of the explanation generation for some benchmark problems, including the time taken for generating the explanations and the size of generated explanations. Our explanation generation system integrates calls to Fast-Downward [53] for planning, VAL [32] for plan validation, and pyperplan [54] for parsing. The results reported here are from experiments run on a 12 core Intel(R) Xeon(R) CPU with an E5-2643 v3@t3.40GHz processor and a 64G RAM. We use three popular planning domains [55] – BlocksWorld, Logistics and Rover – for our experiments. In order to generate explanations we created the human model by randomly removing parts (preconditions and effects) of the action model (the number of edits made per domain is equal to the model patch explanations, which is reported in Table 2). Though the following experiments are only pertinent to action model differences, it does not make any difference at all to the approaches, given the way the state was defined. Also note that these removals, as well as the corresponding model space search, were done in the lifted representation of the domain.

The experimental results are divided into two sections – we first look at the empirical properties of MCEs and MMEs from Section 2 and then at conformant, conditional, and anytime explanations from Section 3.1.

4.1. MCEs versus MMEs

The first item of consideration is the size of explanations with respect to the total number of model differences, since we aimed for minimality as a desired feature for both MCEs and MMEs. Table 2 shows the number of explanations produced and the time taken (in secs) to produce them, against the ground truth. Heuristics seem to provide advantage in terms of the time spent on each problem, particularly for BlocksWorld domain. Further, note how close the approximate version of MCEs are to the exact solutions. As expected, MME search is significantly costlier to compute than MCE. However, note that both

MCEs and MMEs are *significantly smaller* in size ($\sim 20\%$) than the total model difference (which can be arbitrarily large) in certain domains, further underlining the usefulness of generating minimally complete explanations as opposed to dumping the entire model difference on the human. A general rule of thumb is –

$$| \mathcal{E}^{\text{ approx.}MCE} | \leq | \mathcal{E}^{MCE} | < | \mathcal{E}^{MME} | << |\mathcal{E}^{MPE}|$$

Note that the time required to calculate an MME in the Logistics problems is lower than that for the corresponding MCE. This is because for most of these problems a single change in the planner's model made the plan be no longer optimal so that the search ended after checking all possible unit changes. In general, the closer an MCE is to the total number of changes shorter the MME search would be. Also note how PPE solutions, though much easier to compute, do not have completeness and monotonicity properties, and yet often spans the entire model difference, containing information that are not needed to support the optimality of the given plan.

We now increase the number of changes in the human model in BlocksWorld, and illustrate the relative time (in secs) taken to search for exact MCEs in Table 3. The human models are again generated by randomly removing model components (the generated models are not the same as the ones Table 2). As expected there is an exponential increase in the time taken, which can be problematic with even a modest number of model differences. This further highlights the importance of approximations in the model reconciliation process and motivates further research in heuristics for model space search.

Finally, Table 4 illustrates how Proposition 3 reduces the number of nodes searched to find MMEs in random problems from the BlocksWorld domain with 10 faults in the human model, as opposed to the total possible 2^{10} models that can be evaluated – equal to the cardinality of the power set of model changes $|\mathcal{P}(\Gamma(\mathcal{M}^R)\Delta\Gamma(\mathcal{M}_h^R))|$ between the robot model and the mental model.

Problem Instance		MPE (truth)		PPE		MME (exact)		MCE (exact w/o heuristic)		MCE (exact with heuristic)		MCE (approximate)	
		size	time	size	time	size	time	size	time	size	time	size	time
Blocks World	p1	10	n/a	5	n/a	3	1100.8	2	34.7	2	18.9	2	19.8
	p2			8		4	585.9	3	178.4	3	126.6	3	118.8
	p3			4		5	305.3	2	34.7	2	11.7	2	11.7
	p4			7		5	308.6	3	168.3	3	73.3	3	73.0
Rover	p1	10	n/a	10	n/a	2	2093.2	2	111.3	2	100.9	2	101.0
	p2			10		2	2018.4	2	108.6	2	101.7	2	102.7
	p3			10		2	2102.4	2	104.4	2	104.9	2	102.5
	p4			9		1	3801.3	1	13.5	1	12.8	1	12.5
Logistics	p1	5	n/a	5	n/a	4	13.7	4	73.2	4	73.5	4	63.6
	p2			5		4	13.5	4	73.5	4	71.4	4	63.3
	p3			5		5	8.6	5	97.9	5	100.4	3	36.4
	p4			5		5	8.7	5	99.2	5	95.4	3	36.4

Table 2: Comparison of MCEs and MMEs. The size of the explanation corresponds to the cardinality of the explanations (i.e. $\mid \mathcal{E}^* \mid$)

$\mid \mathcal{M}^R \Delta \mathcal{M}_h^R \mid$	problem-1	problem-2	problem-3	problem-4
3	2.2	18.2	4.7	18.5
5	6.0	109.4	15.4	110.2
7	7.3	600.1	23.3	606.8
10	48.4	6849.9	264.2	6803.6

Table 3: MCE search time for increasing model differences for blocksworld.

BlocksWorld	problem-1	problem-2	problem-3	problem-4
Number of nodes expanded for MME (out of 1024)	128	64	32	32

Table 4: Usefulness of Proposition 3 in pruning MME search.

4.2. Conformant, Conditional, and Anytime Explanations

To evaluate explanations against a set of mental models, for each domain, we chose ten problems (generated from the IPC problem generators), and created a new domain and problem pair by removing five random predicates. This new domain and problem represent the ground truth human model. Next, we generate the uncertain estimate of this model by moving three random predicates

into the annotated list. By doing this, we ensure that the ground truth model remains in the completion list of this incomplete model. For these tests, we assume all the possible conditions are equally likely.

As before, Table 5 documents the runtime and the size of explanations generated by each of the algorithms. Note that the `MEGA*-Conditional` was run with γ set to 0.4 and the results for the anytime algorithm only presents the time and size of the *first* solution found. Also, both `MEGA*-Conditional` and `MEGA*-Anytime` expect that it can query the human about the ground truth (each question that the algorithm comes up with is tested against that ground model). The "Question Size" column reports the number of questions that were produced by the search, where each question is related to a single annotated condition, while the "Explanation Size" is the size of the actual explanation presented to the human. For `MEGA*-Conditional` and `MEGA*-Anytime`, we also report the sum of 'Question Size" and "Explanation Size" in parantheses in the explanation column reflecting the total interaction overhead on the human's end.

Unlike `MEGA*-Conditional` and `MEGA*-Anytime`, `MEGA*-Conformant` generates no questions but may produce superfluous explanations. Thus, in the "Explanation Size" column for `MEGA*-Conformant`, we present both the size of the non-superfluous component of the explanation (model updates involving only the known conditions) and the total size of the explanation generated (within parenthesis). The results closely follow intuition. `MEGA*-Anytime` generally takes less time. But since `MEGA*-Anytime` uses a depth first search we cannot guarantee the quality of the solution. In fact, for more than ten problems the solution generated by `MEGA*-Anytime` is strictly worse (in terms Question size + Explanation size) than `MEGA*-Conditional` and for the majority of problems the Question size + Explanation size produced by `MEGA*-Anytime` is strictly larger than the total size of explanations generated by `MEGA*-Conformant`.

Depending on the order in which the successors are visited `MEGA*-Anytime` can end up with smaller sequences. While `MEGA*-Conformant` tend to terminate faster than `MEGA*-Conditional`, the latter produces shorter explanations whenever possible.

Problem Instance		Conformant explanations		Conditional Explanations			Anytime Explanations		
		Explanation Size	Time (secs)	Question Size	Explanation Size	Time (secs)	Question Size	Explanation Size	Time (secs)
Blocksworld	p1	3 (6)	134.84	3	5 (8)	140.75	3	2 (5)	23.7
	p2	1 (1)	1.64	0	1 (1)	9.2	0	2 (2)	7.32
	p3	2 (3)	20.56	1	3 (4)	55.91	3	3 (6)	20.51
	p4	1 (2)	11.23	1	2 (3)	128.5	0	9 (9)	21.51
	p5	3 (6)	130.64	3	5 (8)	150.61	3	3 (6)	29.43
	p6	2 (4)	279.71	2	4 (6)	539.2	3	2 (5)	25.78
	p7	2 (5)	343.04	3	1 (4)	495.2	3	3 (6)	26.79
	p8	3 (3)	60.35	0	3 (3)	204.72	0	3 (3)	9.7
	p9	2 (4)	234.7	2	4 (6)	379.21	2	2 (4)	18.57
	p10	1 (3)	218.38	3	2 (5)	444.61	2	2 (4)	19.92
Logistics	p1	2 (4)	62.3	2	4 (6)	99.78	2	2 (4)	21.96
	p2	2 (5)	61.45	3	5 (8)	80.73	3	3 (6)	26.68
	p3	3 (5)	246.23	2	4 (6)	297.71	2	2 (4)	21.6
	p4	2 (5)	54.79	3	5 (8)	72.69	3	3 (6)	21.63
	p5	2 (5)	59.87	3	5 (8)	86.72	3	3 (6)	26.04
	p6	2 (4)	489.36	2	3 (5)	729.42	3	2 (5)	24.54
	p7	2 (5)	402.66	1	2 (3)	544.23	3	3 (6)	28.98
	p8	3 (5)	522.47	2	4 (6)	731.1	3	3 (6)	17.78
	p9	3 (6)	1719.26	3	4 (7)	1535.02	3	1 (4)	22.48
	p10	4 (6)	1747.62	2	5 (7)	1783.33	2	4 (6)	21.2
Rover	p1	2 (2)	3.83	0	1 (1)	8.63	0	3 (3)	9.37
	p2	2 (3)	26.93	1	2 (3)	141.2	2	3 (5)	12.91
	p3	2 (4)	99.02	2	3 (5)	165.82	3	2 (5)	25.62
	p4	3 (4)	102.57	1	3 (4)	253.41	1	4 (5)	13.57
	p5	1 (2)	14.87	0	1 (1)	10.58	3	2 (5)	25.65
	p6	1 (2)	146.21	1	1 (2)	835.16	1	4 (5)	14.15
	p7	2 (3)	182.81	1	2 (3)	599.48	1	3 (4)	15.31
	p8	1 (1)	12.07	0	1 (1)	32.92	0	1 (1)	5.14
	p9	1 (2)	125.49	1	2 (3)	523.48	1	1 (2)	15.74
	p10	1 (2)	89.89	1	2 (3)	525.24	2	1 (3)	19.57

Table 5: Runtime and solution size for explanations with uncertain mental models.

# of models →	2	4	8	16
Baseline	10.95	41.71	195.81	936.30
`MEGA*-Conformant`	11.11	37.01	117.26	291.88

Table 6: Comparison of the runtime for `MEGA*-Conformant` versus the time needed to run MCE for every member of the completion set.

Finally, the purpose of compiling the set of possible models into $\mathcal{M}_{max}$ and $\mathcal{M}_{min}$ is that we no longer need to compute explanations over each individual model in the set of possible models separately (baseline). Table 6 illustrates the significant scale-ups we can achieve as a result of this.

5. Human-Factors Study of the Model Reconciliation Process

The design of explainable AI algorithms is, of course, incomplete without evaluations with actual humans in the loop. Thus, in the following discussion, we will report on the the salient findings from a series of controlled user studies we undertook in order to evaluate the usefulness of the model reconciliation approach. Through these studies, we aim to validate whether explanations in the form of model reconciliation (in its various forms) suffice to explain the optimality and correctness of plans to the human in the loop. We also study participants who were asked to generate explanations in the form of model changes, to see if explanations generated by the humans align with any of the multi-model explanations identified in the discussion so far. The studies suggest that humans do indeed understand explanations of this form and believe that such explanations are necessary to explain plans.

We stick to the USAR domain for our study (Figure 9). In the study, we only simulate the interface to the external. As we discussed before, in general, differences in the models of the human and the robot can manifest in any form (e.g. the robot may have lost some capability or its goals may have changed). In the current setup, however, we only deal with differences in the map of the environment as available to the two agents. Note that we only evaluate explanations types from Section 2 in the user study: those in Section 3.1 have identical properties (conformant explanations are MCEs while contingent explanations are conformant explanations after reducing uncertainty over the mental model) differing only in the estimation of the mental model which is a learning problem outside of the scope of this paper.

5.1. Study – 1: Participants are explainers

The first part of the study aims to develop an understanding of how humans respond to the task of generating explanations, i.e. if left to themselves, humans preferred to generate explanations similar to the ones developed in this paper. To test this, we asked participants to assume the role of the internal agent in

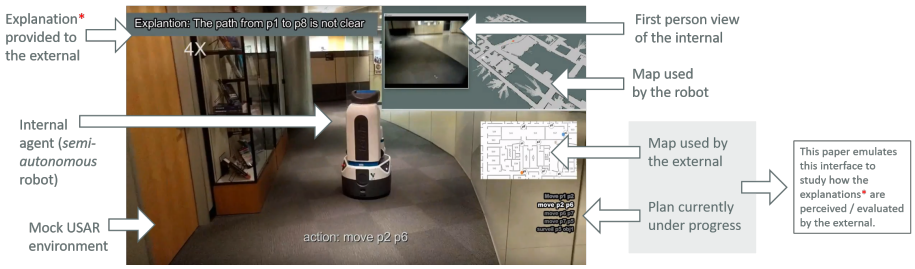


Figure 9: Illustration of the simulated USAR setting. We expose a mock interface to the external agent (right inset) on the browser to study the properties of different explanations afforded by the model reconciliation framework.

the explanation process and explain their plans with respect to the faulty map of their teammate. Specifically, we set out to test the following hypothesis –

H1. When asked to, participants would leverage model differences as a key ingredient for explanations.

H1a. Explanation generated by participants would demonstrate contrastiveness. Thus, PPE type explanations would be overlooked in favor of complete solutions (MCEs and MPEs) when there are multiple competing hypothesis for the human.

H2. Participants would like to minimize the content of the explanation by removing details that are not relevant to the plan being explained.

H2a. Explanations from participants would be closer to MCEs than MPEs.

H2b. This should be even more significant if restrictions are placed on communication.

As a result of this study, we intend to identify to what extent explanation types developed in this paper built upon principles of explanations in human-human interactions studied in social sciences (more on this in Section 6) truly reflect human intuition.

Note that we primed the subjects to annotate changes in the map, while giving them the opportunity to –

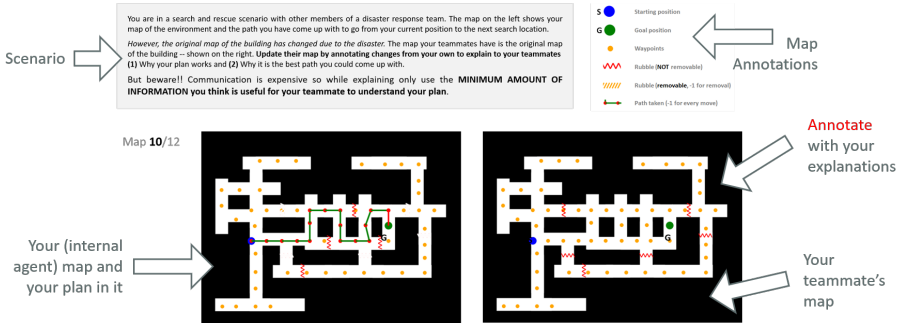


Figure 10: Interface for Study-1: Participants assumed the role of the internal agent and explained their plans to a teammate with a possibly different map of the world.

1. Provide more than annotations (and we did find other interesting kinds of explanations emerge as we discuss later in Section 5.3)

2. Comment on the sufficiency and necessity of such explanations (as we report in Section 5.1.2)

The reason for this choice is because in the work being evaluated here, communicating model differences has been considered to be the starting point of the explanation process. So we start from that assumption and evaluate to what extent the kinds of explanations introduced here – MCE / MPE / PPE / etc. – are actually useful. Additionally, this setup also helps to re-contextualize the real importance of model difference in the explanation process in light of reasons explained in (1) and (2) above.

5.1.1. Experimental Setup

Figure 10 shows an example map and plan provided to a participant. On the left side, the participant is shown the actual map along with their plan, starting position and goal. The panel on the right shows the map available to the explainee. The maps have removable and non-removable rubble blocking access to certain paths (the maps may disagree as to the locations of the debris). The participants were asked to convince the explainee of the correctness and optimality of the given plan by updating the latter's maps with annotations they

felt were relevant in achieving that goal. We ran the study with two conditions –

C1. Here the participants were asked to ensure, via explanations, that their plan was correct and optimal in the updated model of their teammate;

C2. Here, in addition to C1, they were also asked to use the minimal amount of information they felt was needed to achieve the condition in C1.

Each participant was shown how to annotate (not an explanation) a sample map and was then asked to explain 12 different plans using similar annotations. After each participant was finished with their assignment, they were asked the following subjective questions –

`Q1. Providing map updates were necessary to explain my plans.`
`Q2. Providing map updates were sufficient to explain my plans.`
`Q3. I found that my plans were easy to explain.`

The answers to these questions were measured using a five-point Likert scale. The answers to the first two questions will help to establish whether humans considered map updates (or in general updates on the model differences) at all necessary and/or sufficient to explain a given plan. The final question measures whether the participants found the explanation process using model differences tractable. It is important to note that in this setting we do not measure the efficacy of these explanations (this is the subject of Study-2 in Section 5.2). Rather we are trying to find whether a human explainer would have naturally participated in the model reconciliation approach during the explanation process.

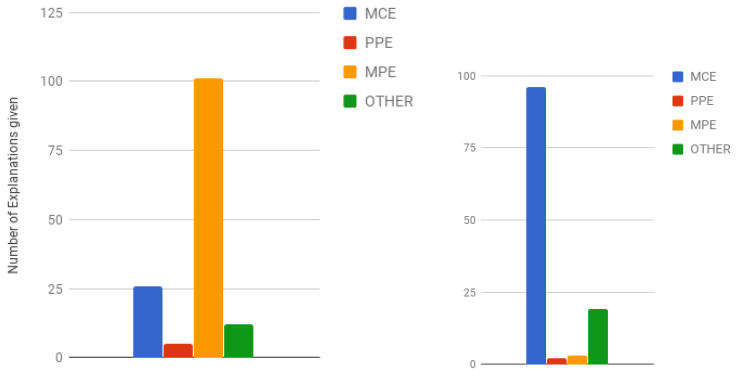
In total, we had 12 participants for condition C1 and 10 participants for condition C2 including 7 female and 18 male participants between the age range of 18-29 (data corresponding to 5 participants who misinterpreted the instructions had to be removed, 2 participants did not reveal their demographics). Participants for the study were recruited by requesting the department secretary to send an email to the student body to ensure that they had no prior knowledge about the study or its relevance. Each participant was paid $10.

5.1.2. Results

Figure 11 – The first hypothesis we tested was whether the explanations generated by the participants matched any of the explanation types introduced in this paper. We did this by going through all the individual explanations provided by the participants and then categorizing each explanation to one of the four types, namely MCE, PPE, MPE or Other (the "other" group contains explanations that do not correspond to any of the predefined explanation types – more on this later in Section 5.3). Each explanation type was identified by checking the explanation provided by the participants against what may have been generated by the algorithm. Figure 11a shows the number of explanations of each type that were provided by the participants of C1. The graph shows a clear preference for MPE, i.e. providing all model differences. A possible reason for this may be since the size of MPEs for the given maps were not too large (and participants did not have time constraints). Interestingly, in C2 we see a clear shift in preferences (Figure 11b) where most participants ended up generating MCE style explanations. This means at least for scenarios where there are constraints on communication, the humans would prefer generating MCEs as opposed to explaining all the model differences.

These findings are consistent with H1, with very few of the explanations in type "Other" (Figure 11). This is also backed up by answers to subjective questions Q1 and Q2 above. Further, the preference of MPE/MCE over PPE (H1a) is quite stark. Contrary to H2a, participants seemed to have preferred full model explanation (MPE) in C1 condition which is surprising. However, results of C2 condition are more aligned with H2b, even though we expected to see a similar trend (if not as strong) in C1 condition as well.

Figures 12 and 13 – These show the results of the subjective questions for C1 and C2 respectively. Interestingly, in C1, while most people agreed on the necessity of explanations in the form of model differences, they were less confident regarding the sufficiency of such explanations. In fact, we found that many participants left additional explanations in their worksheet in the form of



(a) Study-1:C1 (b) Study-1:C2

Figure 11: Explanation counts for Study-1:C:1-2.

free text (we discuss some of these findings in Section 5.3). In C2, we still see that more people are convinced about the necessity of these explanations than sufficiency. But we see a reduction in the confidence of the participants, which may have been caused by the additional minimization constraints.

5.2. Study – 2: Participants are explainees

Now we study how different kinds of explanations outlined in Section 2 are perceived by the participants. This study was designed to provide clues to how humans comprehend explanations when provided to them in the form of model differences. Specifically, we intend to evaluate the following hypothesis, in line with the intended properties of each of the explanation types –

H1. Participants would be able to identify optimality given an MPE or MCE.

H2. Participants would be able to identify executability but possible suboptimality of a plan given a PPE.

H3. Participants would not ask for explanations when presented with explicable plans (optimal in mental model).

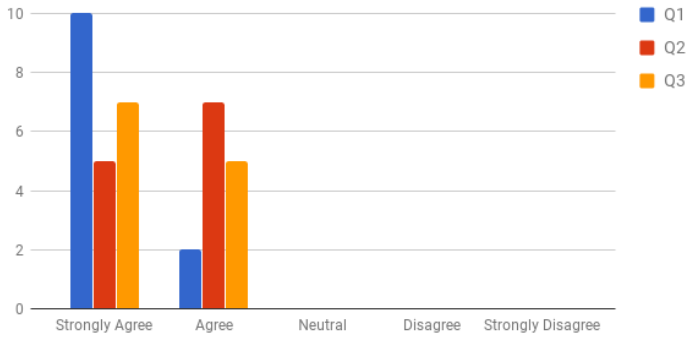


Figure 12: Subjective responses of participants in Study-1:C1.

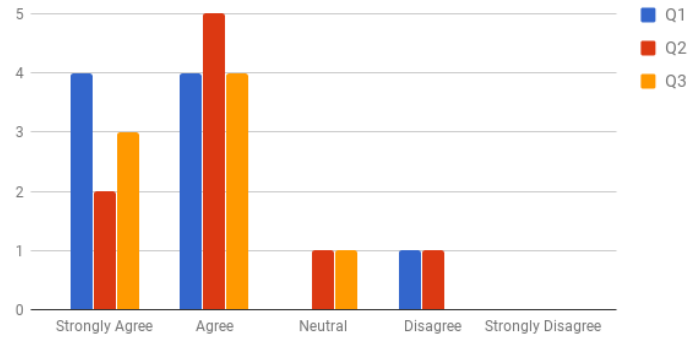


Figure 13: Subjective responses of participants in Study-1:C2.

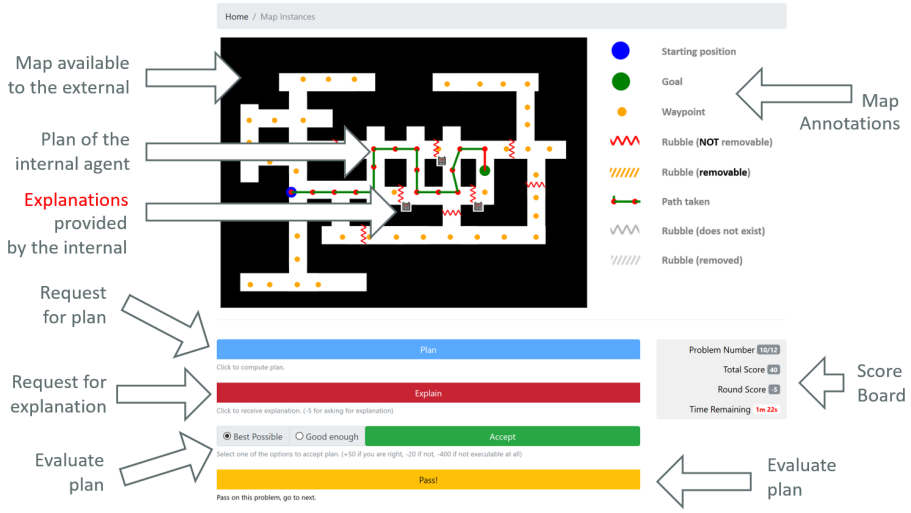


Figure 14: Interface for Study-2 where participants assumed the role of the external commander and evaluated plans provided by the internal robot. They could request for plans and explanations to those plans (if not satisfied) and rate them as optimal or suboptimal or (if unsatisfied) can chose to pass.

As a result of this study, we intend to validate whether desired properties of explanations for task planning designed by following norms and principles outlines in the social sciences in the context of human-human interactions [13] do actually carry over for human-robot interactions.

5.2.1. Experimental Setup

During this study, participants were incentivized to make sure that the explanation does indeed help them understand the optimality and correctness of the plans in question by formulating the interaction in the form of a game.

Figure 14 shows a screenshot of the interface.[10] The game displays to each

[10]This domain has elements of both motion planning and task planning (e.g. removal of debris) in it. The approaches developed in this paper are applicable to task plans in general, as done in the work on identifying preferences over logically equivalent explanations in [33], where the study was conducted in a logistic domain with plans involving the transport of cargo. User studies have also been undertaken to test the validity of many variants of model reconciliation, including a warehouse scenario in [51] as well as logistics and travel scenarios in [52].

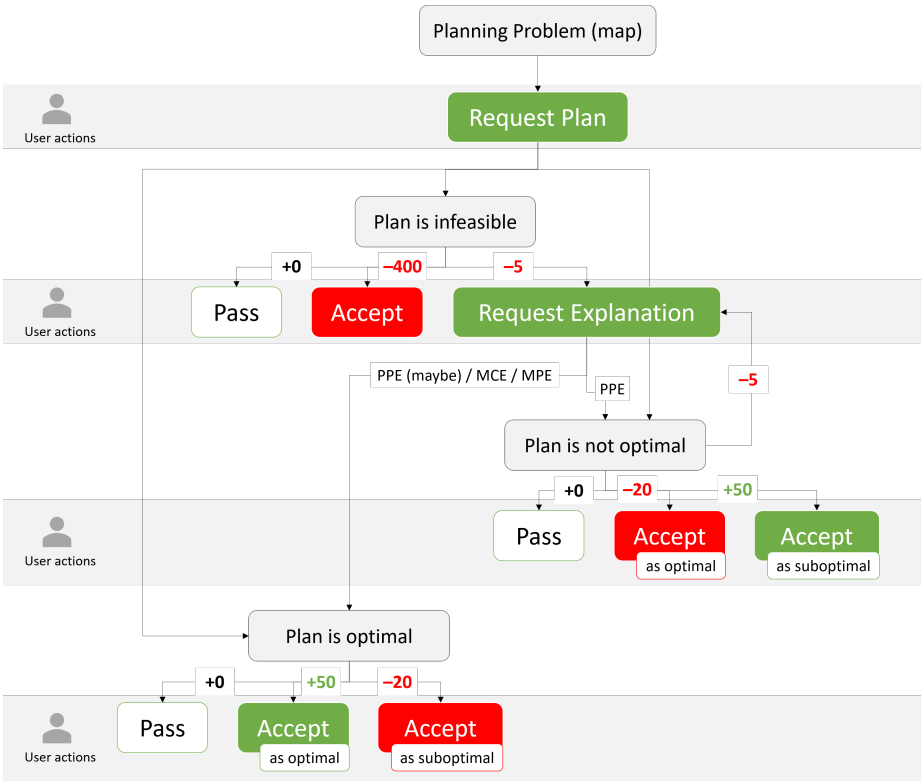


Figure 15: Illustration of the flow of logic in the experimental setup.

participant an initial map (which they are told may differ from the robot's actual map), the starting point and the goal. Once the player asks for a plan, the robot responds with a plan illustrated as a series of paths through waypoints highlighted on the map. The goal of the participant is to identify if the plan shown is optimal or just satisficing. If the player is unsure of the path, they can ask for an explanation from the robot. The explanation is provided to the participant in the form of a set of model changes in the player's map. If the player is still unsure, they can click on the pass button to move to the next map.

The scoring scheme for the game is as follows (Summarized in Figure 15). Each player is awarded 50 points for correctly identifying the plan as either optimal or satisficing. Incorrectly identifying an optimal plan as suboptimal or vice versa would cost them 20 points. Every request for explanation would

further cost them 5 points, while skipping a map does not result in any penalty. The participants were additionally told that selecting an inexecutable plan as either feasible or optimal would result in a penalty of 400 points. Even though there were no actual incorrect plans in the dataset, this information was provided to deter participants from taking chances with plans they did not understand well.

Each participant was paid $10 dollars and received additional bonuses based on the following payment scheme –

- Scores higher than or equal to 540 were paid $10.
- Scores between 540 and 440 were paid $7.
- Scores between 440 and 340 were paid $5.
- Scores between 340 and 240 were paid $3.
- Scores below 240 received no bonuses.

The scoring systems for the game was designed to ensure

- Participants should only ask for an explanation when they are unsure about the quality of the plan (due to small negative points on explanations).
- Participants are incentivized to identify the feasibility and optimality of the given plan correctly (large reward and penalty on doing this wrongly).

Each participant was shown a total of 12 maps (same maps as in Study–1). For 6 of the 12 maps, the player was assigned the optimal robot plan, and when they asked for an explanation, they were randomly shown either MCE, PPE or MPE explanation with regards to the robot model. For the rest of the maps, participants could potentially be assigned a plan that is optimal in the human model (i.e. an explicable plan) or somewhere in between as introduced in [56] (referred to henceforth as the balanced plan) in place of the robot optimal plan[11]. The participants that were assigned the optimal robot plan were provided

[11] Note that of the 6 maps, only 3 had both balanced as well as explicable plans, the rest either

an MCE, PPE or MPE explanation, otherwise they were provided an empty explanation for the explicable plan. Also note that for 4 out of the 12 maps the PPE explanation cannot prove the optimality of the plan.

At the end of the study, each participant was presented with a series of subjective questions as follows. The responses to each question were measured on a five-point Likert scale.

```
Q1. The explanations provided by the robot was helpful.
Q2. The explanations provided by the robot was easy to understand.
Q3. I was satisfied with the explanations.
Q4. I trust the robot to work on its own.
Q5. My trust in the robot increased during the study.
```

In total, we had 27 participants for Study–2, including 4 female and 22 male between the ages of 19 to 31 (1 participant did not reveal their demographic).

5.2.2. Results

Figure 16 – As we mentioned before, the goal of this study is to identify if explanations in the form of model reconciliation can convey to humans the optimality and correctness of plans. Here, each participant was shown the 12 maps from Study-1 and each map was assigned a random explanation type (and in some cases different plans). We wanted to identify whether the participants that asked for explanations were able to come up with the correct conclusions. We chose to focus on participants who decided to ask for explanations, as people who didn't request for one may be operating off of a model different from the presented one. If this was indeed the case, results collected from these participants will not match the assumption required for model reconciliation

had a balanced plan or the optimal human plan. Note that balanced plans are indistinguishable from the optimal plan from the point of view of the human. They are more useful to the robot for trading of explanation and explicability costs. Hence, we did not expand on further results on balanced plans here so as not to distract from the main focus of the paper which is to evaluate explanations as model reconciliation. A detailed treatise is available in [56].

explanations.. This means that the subjects who asked for MCE and MPE were able to correctly identify the plans as optimal, while the people who received PPE were able to correctly classify the plan to either optimal or satisficing (i.e. for all but 5 maps PPE is enough to prove optimality).

Figure 16 shows the statistics of the selections made by participants who had requested an explanation. The right side inset shows the percentage (for every map instance) of participants who selected the correct options (blue), the incorrect ones (red) or simply passed (orange), while the left side shows the average across all 12 maps. We notice that in general people were overwhelmingly able to identify the correct choice. Even in the case of PPEs, where the explanations only ensured correctness (map instances 1, 2, 3, 8 and 11) the participants were able to make the right choice. This is consistent with H1 and H2 and demonstrates that explanations in the form of model reconciliation are a viable means of conveying the correctness and optimality of plans – i.e. participants can differentiate between completeness and incompleteness of explanations.

Figure 17 – These conclusions are further supported by results from the subjective questionnaire (Figure 17). Most people seem to agree that the explanations were helpful and easy to understand. In fact, the majority of people strongly agreed that their trust of the robot increased during the study.

Figure 18 – We were also curious (H3) about the usefulness of explicable plans (that are optimal in human's model), i.e. if the subjects still asked for explanations when presented with explicable plans. Figure 18 shows the percentage of times subjects asked for explanations when presented with explicable versus robot optimal plans. The rate of explanations is considerably less in case of explicable plans as hypothesized. This matches the intuition behind the notion of plan explicability as a viable means (in addition to explanations) of dealing with model divergence in human-in-the-loop operation of robots.

It is interesting to see that in Figure 18 about a third of the time participants still asked for explanations even when the plan was explicable, and thus optimal in their map. We believe this is an artifact of the risk-averse behavior incentivized

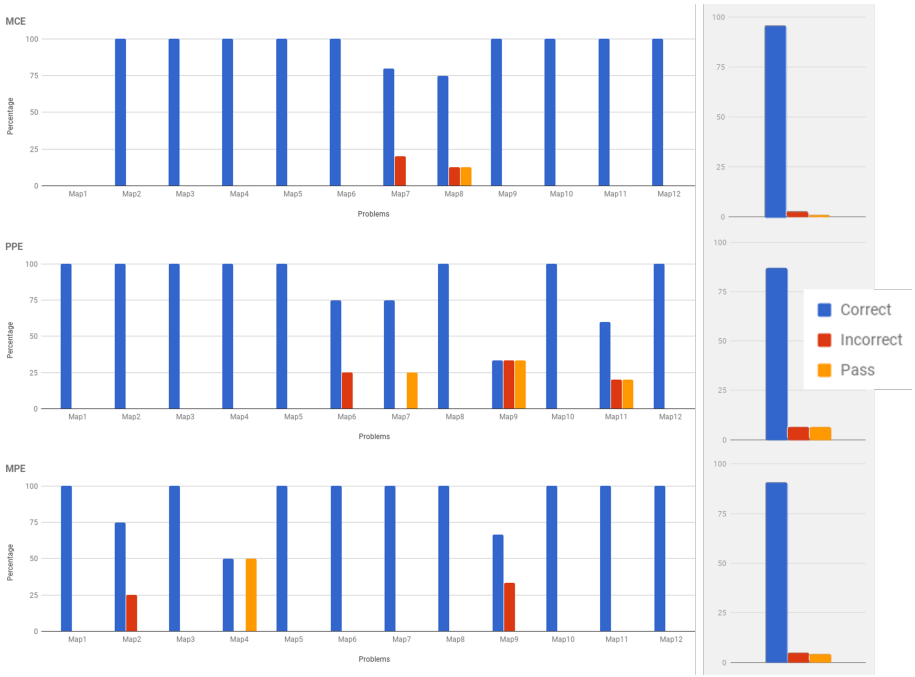


Figure 16: Percentage of times different explanations (i.e. MCE / MPE / PPE) led to correct decision on the human's part in each problem (the aggregated result is shown on the right). A "correct decision" involves recognizing optimality of the robot plan on being presented an MCE or MPE, and optimality or executability (as the case may be) in case of a PPE.

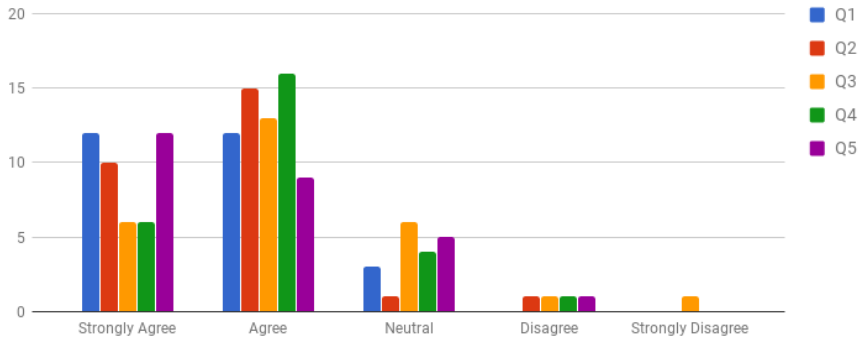


Figure 17: Subjective responses of participants in Study–2.

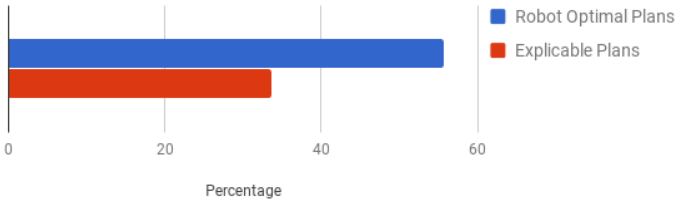


Figure 18: Percentage of times explanations were sought for in Study–2 when participants presented with explicable plans versus robot optimal plans with explanations.

by the gamification of the explanation process. This is to make sure that participants were sufficiently invested in the outcome as well as mimic the high-stakes nature of USAR settings to accurately evaluate the explanations. It is also an indication of the cognitive burden on the humans who may not be (cost) optimal planners. While this is consistent with the spirit of H3, the finding is also somewhat indicative of the limitations of plan explicability as it is defined in existing literature at the moment [56]. Thus, going forward, the objective function should incorporate the cost or difficulty of analyzing the plans and explanations from the point of view of the human in addition to the current costs of explicability and explanations modeled from the perspective of the robot.

Interestingly, the participants also did not ask for explanations around 40% of the time (c.f. Figure 18) when they "should have" (i.e. suboptimal plan in the human model) according to the theory of model reconciliation. We noticed no clear trend here (e.g. decreasing rate for explanations asked due to increasing

		Outcome	Comments
Study-1	H1	✓	Participants largely agreed that model reconciliation was a necessary and sufficient part of the explanation process.
	H1a	✓	Participants preferred explanations that are complete, and preserve contrastive property across multiple hypothesis.
	H2	✗	Participants did not care to minimize size of explanations, i.e. exclude irrelevant details.
	H2a	✗	Explanations generated by participants in the free form condition were largely of the form of MPEs.
	H2b	✓	Participants did generate MCEs when their communication capability was explicitly restricted.
Study-2	H1	✓	Participants could identify the optimality of the given plan with complete explanations.
	H2	✓	Participants could identify suboptimality of the given plan for incomplete explanations.
	H3	✓ / ?	Some participants asked for explanations even for explicable plans, though the majority did not.

Table 7: Summary of results from the user studies.

trust). This was most likely due to limitations of inferential capability of humans and a limitation of the existing formulation of model reconciliation as well. The overall results from the study are also summarized in Table 7.

5.3. Discussion: Other kinds of explanations

As we mentioned before, there were some instances where the participants from Study 1 generated explanations that are outside the scope of any of the explanation types discussed in Section 2.2. These were marked as "Other" in Figure 11. In the following, we discuss three cases that we found interesting.

Post-hoc explanations. Notice that parts of an MCE that contribute to the executability of a plan need not be explained in situations where the robot is explaining plans that have *already been done* as opposed to those that are being proposed for execution. The rationale behind this is that if the human sees an action, that would not have succeeded in his model, actually end up succeeding (e.g. the robot had managed to go through a corridor that was blocked by rubble) then he can rationalize that event by updating his own model (e.g. there must not have been a rubble there). This seems to be a viable approach to further reduce size (c.f. selective property of explanations in [13]) of explanations in a post-hoc setting, and is out of scope of explanations developed here.

Identification of Explicit Foils. Identification of explicit foils can help reduce the size of explanations as well. In the explanations introduced in Section 2 the foil was implicit – i.e. why this plan *as opposed to all other plans*. However, when the implicit foil can be estimated (e.g. top-K plans expected by the human or in estimation of the mental model from the foil as done in [45]) then the explanations can only include information on why the plan in question is better than those other options (which are either not executable or costlier). Some participants provided explanations contrasting some of these foils in terms of (and in addition to just) the model differences.

Cost-based reasoning. Finally, a kind of explanation that was attempted by some participants involved a cost analysis of the current plan with respect to foils (in addition to model differences, as mentioned above). Such explanations have been studied extensively in previous planning literature [10, 11] and is still relevant for plan explanations on top of the model reconciliation process.

6. Related Work

We started out in the introduction with the premise that plan explanations cannot be a soliloquy but is rather a means of reconciling differences between the AI model and the user expectations or the mental model of the user, thereby

establishing common grounds with the human in the loop. [57] Much of the work we cited there assume that the model of the planner and the end user are the same. This does not bear out in many applications and we saw some examples of this above. While we referred to relevant work on that topic in the course of our presentation wherever necessary, for a more detailed treaties of the evolution of the world of explainable AI planning, we refer the reader to [2].

One particular work we want to expand on a bit more here is a recent survey on lessons learned from social sciences on the dynamics of the explanation process in human-human interactions. [13] The work outlines three key properties of explanations – *social* (in being able to model the explainee's expectations), *contrastiveness* (the ability to contrast potential foils), and *selectiveness* (to prioritize model details for explanations). Our approach is inherently social (by explicitly accounting for the mental model of the explainee). We also spent a fair bit of time expounding on the contrastive property in the paper, while our method of selection is determined by the minimality and monotonicity criterion.[12] While the contrastive property has been the subject of much interest in the explainable AI planning community of late [59, 60], to the best of our knowledge, the model reconciliation process remains the only existing plan explanation process that conforms to all three properties of social, contrastiveness, and selectiveness of explanations, as outlined in [13].

Our view of explanation as a model reconciliation process is further supported by studies in the field of psychology which stipulate that –

> *"... explanations privilege a subset of beliefs, excluding possibilities inconsistent with those beliefs... can serve as a source of constraint in reasoning..." [61]*

This is achieved in our case by the appropriate change in the expectation of

[12] As we mentioned before, since minimal explanations in the model reconciliation framework are not unique, the selectiveness criterion can be further explored [33] in the context of preferences over logically equivalent explanations. Recent work exploring the representation of plan properties for purposes of explanation [58] can also help in this cause.

the model that is believed to have engendered the plan in question. Furthermore, authors in [62] also underline that –

> *"... explanations are typically contrastive... the contrast provides a constraint on what should figure in a selected explanation..." [62]*

This is especially relevant in order for an explanation to be self-contained and unambiguous. Hence the requirement of optimality in our explanations, which not only ensures that the current plan is valid in the updated model, but is also better than other alternatives. This is consistent with the notion of optimal (single-model) explanations investigated in [5] where less costly plans are referred to as *preferred explanations*. The optimality criterion, and argumentation over the human mental model, makes the problem fundamentally different from model change algorithms in [63, 64, 65, 36, 66] which focus more on the feasibility of plans or correctness of domains, or tackle model extensions in general without consideration of the human expectations i.e. the human mental model, and the constrastive property of potential foils in it.

The field of epistemic reasoning is also closely related to model reconciliation explanations as studied within this paper. In fact, works like [67], have already used the epistemic reasoning framework to generalize model reconciliation beyond just classical planning problems. In [68] we also leveraged tools from epistemic planning [69, 70] to incorporate the reasoning about model reconciliation explanations into the planning process through the notion of "explanatory actions" or robot actions with purely epistemic effects that result in the update of the human's belief regarding the robot model.

7. Concluding Remarks

This concludes a comprehensive account of the model reconciliation framework as a means of formalizing the explanation process of a decision making problem in terms of the differences between the agent model and the mental model of the explainee. We started with the basic framework, showed how to relax

assumptions to make it more palatable to the real world, and evaluated the properties of explanations in the model reconciliation framework with empirical evaluations as well as controlled user studies. We will end with a few pointers to applications where these approaches have already found use, and a brief description of future work.

7.1. Applications

In the course of discussion, we used two illustrative domains to demonstrate the different aspects of the model reconciliation process. The first was motivated by the anecdotal evidence of consequences of not accounting for mental mdoel of users (Figure 2) while the latter is modeled after a real-world USAR domain [15] we use in our inter-disciplinary collaborations with our human factors team at Arizona State University. [71, 72]

In the wider world, the concept of model reconciliation has seen intriguing deployments of explainable AI systems. One such application was seen in the then Cognitive Environments Laboratory (CEL) in the IBM T.J. Watson Research Lab in Yorktown where the concept of model reconciliation was used to establish common grounds between an embodied assistant and the inhabitants of a smart room environment. [73] This is an especially interesting setting since for embodied agents in smart rooms, it is especially hard for people interacting with it to know what it sees, what it hears, and what the state of its model is.

Another application of model reconciliation in the industry has been in assistance of domain authors for dialogue planning. [74] Here, the role of the explainable agent is flipped from explaining its own plans to explaining differences in the desired model to be authors versus the current one. It is interesting to see such varied roles of an explainable AI planning agent in other applications as well – in [26] we looked at how the planner can act as decision support and explain plan recommendations to a human planner. In a special case of decision support, we demonstrated in [75] how principles of model reconciliation can be adopted to model students and generate curriculum for an intelligent tutoring system Dragoon [76] deployed at Arizona State University. We presented some of

these applications at the recently concluded AAAI 2020 Tutorial on Synthesizing Explainable and Deceptive Behavior for Human-AI Interaction. [77]

7.2. Future Work

An area of active research which we left mostly untouched is that of learning of mental models. We mentioned works specific to model learning in the model reconciliation framework, where the mental model is estimated from the foils [45] or preferred explanations are learned in the course of interactions [51]. This is a broader area of research, ranging from learning of causal relationships for explanations [78] to the learning of mental models iterative [79, 80] in the presence of uncertainty in preferences.

Interestingly, the notion of explanation as model reconciliation has already been adopted beyond planning formalisms, such as in the context of explaining logical programs [48, 81]. There is also a growing consensus in the need to consider the knowledge content of the mental model of humans in related activities like plan recognition [67] that, in general, allow the planner to empathize with human teammates [82]. These are certainly interesting developments in the broader world of model reconciliation and mental models in planning.

Finally, in this paper, we further focused mostly on the generation of the *content* of explanations rather than the actual delivery of that information. Depending on the type of interaction between the planner and the human, this can be achieved by means of natural language dialog [83], in the form of a graphical user interface [26, 73] or even in mixed-reality interfaces [84]. The question of explainable AI and user interaction is indeed inseparable [85] and is going to be a topic of great interest going forward as AI planning techniques mature and interface with end users.

Acknowledgements. Kambhampati's research is supported in part by ONR grants N00014-16-1-2892, N00014-18-1-2442, N00014-18-1-2840, N00014-19-1-2119, AFOSR grant FA9550-18-1-0067, DARPA SAIL-ON grant W911NF-19-2-0006, NSF grants 1936997 (C-ACCEL), 1844325, and a NASA grant

NNX17AD06G. Chakraborti was also supported by the IBM Ph.D. Fellowship during the formative years of the project.

A special word of thanks to Sachin Grover (Arizona State University) for his help in setting up the user studies, and to Prof. Yu Zhang (Arizona State University) for his helpful ideas and suggestion in the early stages of the work.

References

[1] D. S. Weld, G. Bansal, The Challenge of Crafting Intelligible Intelligence, Communications of the ACM (2019).

[2] T. Chakraborti, S. Sreedharan, S. Kambhampati, The Emerging Landscape of Explainable AI Planning and Decision Making, IJCAI (2020).

[3] P. Langley, B. Meadows, M. Sridharan, D. Choi, Explainable Agency for Intelligent Autonomous Systems, in: AAAI/IAAI, 2017.

[4] S. Kambhampati, A Classification of Plan Modification Strategies Based on Coverage and Information Requirements, in: AAAI Spring Symposium on Case Based Reasoning, 1990.

[5] S. Sohrabi, J. A. Baier, S. A. McIlraith, Preferred Explanations: Theory and Generation via Planning, in: AAAI, 2011.

[6] B. Seegebarth, F. Müller, B. Schattenberg, S. Biundo, Making Hybrid Plans More Clear to Human Users – A Formal Approach for Generating Sound Explanations, in: ICAPS, 2012.

[7] B. L. Meadows, P. Langley, M. J. Emery, Seeing Beyond Shadows: Incremental Abductive Reasoning for Plan Understanding, in: AAAI Workshop on Plan, Activity, and Intent Recognition (PAIR), 2013.

[8] P. Langley, Varieties of Explainable Agency, in: ICAPS Workshop on Explainable AI Planning (XAIP), 2019.

[9] Y. Zhou, D. Danks, Different "Intelligibility" for Different Folks, in: AIES/AAAI, 2020.

[10] M. Fox, D. Long, D. Magazzeni, Explainable Planning, in: IJCAI Workshop on Explainable AI (XAI), 2017.

[11] D. E. Smith, Planning as an Iterative Process, in: AAAI, 2012.

[12] M. Cashmore, A. Collins, B. Krarup, S. Krivic, D. Magazzeni, D. Smith, Towards Explainable AI Planning as a Service, in: ICAPS Workshop on Explainable AI Planning (XAIP), 2019.

[13] T. Miller, Explanation in Artificial Intelligence: Insights from the Social Sciences, Artificial Intelligence Journal (2017).

[14] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, D. Wilkins, PDDL – The Planning Domain Definition Language (1998).

[15] C. E. Bartlett, Communication between Teammates in Urban Search and Rescue, Thesis (2015). Arizona State University.

[16] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, A. M. Erkmen, Search and Rescue Robotics, in: Handbook of Robotics, 2008.

[17] Y. Zhang, V. Narayanan, T. Chakraborti, S. Kambhampati, A Human Factors Analysis of Proactive Assistance in Human-Robot Teaming, in: IROS, 2015.

[18] K. Talamadupula, G. Briggs, T. Chakraborti, M. Scheutz, S. Kambhampati, Coordination in Human-Robot Teams Using Mental Modeling and Plan Recognition, in: IROS, 2014.

[19] S. Russell, P. Norvig, Artificial intelligence: a modern approach, Prentice Hall, 2003.

[20] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, S. Kambhampati, Plan Explicability and Predictability for Robot Task Planning, in: ICRA, 2017.

[21] Y. Zhang, Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, S. Kambhampati, Plan Explicability for Robot Task Planning, in: RSS Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics, 2016.

[22] A. Kulkarni, T. Chakraborti, Y. Zha, S. G. Vadlamudi, Y. Zhang, S. Kambhampati, Explicable Robot Planning as Minimizing Distance from Expected Behavior, in: AAMAS Extended Abstract, 2019.

[23] B. Srivastava, T. A. Nguyen, A. Gerevini, S. Kambhampati, M. B. Do, I. Serina, Domain Independent Approaches for Finding Diverse Plans, in: IJCAI, 2007.

[24] T. A. Nguyen, M. Do, A. E. Gerevini, I. Serina, B. Srivastava, S. Kambhampati, Generating Diverse Plans to Handle Unknown and Partially Known User Preferences, Artificial Intelligence Journal (2012).

[25] A. Riabov, S. Sohrabi, O. Udrea, New Algorithms for The Top-K Planning Problem, in: Scheduling and Planning Applications Workshop (SPARK) at ICAPS, 2014.

[26] S. Grover, S. Sengupta, T. Chakraborti, A. P. Mishra, S. Kambhampati, RADAR: Automated Task Planning for Proactive Decision Support, in: HCI Journal, 2020.

[27] T. Chakraborti, S. Kambhampati, (How) Can AI Bots Lie?, in: ICAPS Workshop on Explainable AI Planning (XAIP), 2019.

[28] A. Isaac, W. Bridewell, White Lies on Silver Tongues: Why Robots Need to Deceive (and How), Journal of Robot Ethics (2017).

[29] T. Chakraborti, S. Kambhampati, (When) Can AI Bots Lie?, in: AIES/AAAI, 2019.

[30] J. J. Palmieri, T. A. Stern, Lies in the doctor-patient relationship, Primary care companion to the Journal of clinical psychiatry 11 (2009) 163.

[31] M. Fox, R. Howey, D. Long, Validating Plans in the Context of Processes and Exogenous Events, in: AAAI, 2005.

[32] R. Howey, D. Long, M. Fox, VAL: Automatic Plan Validation, Continuous Effects and Mixed Initiative Planning Using PDDL, in: ICTAI, 2004.

[33] Z. Zahedi, A. Olmo, T. Chakraborti, S. Sreedharan, S. Kambhampati, Towards Understanding User Preferences for Explanation Types in Explanation as Model Reconciliation, in: HRI, 2019. Late Breaking Report.

[34] C. Wayllace, P. Hou, W. Yeoh, T. C. Son, Goal Recognition Design with Stochastic Agent Action Outcomes, in: IJCAI, 2016.

[35] S. Keren, L. Pineda, A. Gal, E. Karpas, S. Zilberstein, Equi-reward Utility Maximizing Design in Stochastic Environments, in: IJCAI, 2017.

[36] D. Bryce, J. Benton, M. W. Boldt, Maintaining Evolving Domain Models, in: IJCAI, 2016.

[37] T. Nguyen, S. Sreedharan, S. Kambhampati, Robust Planning with Incomplete Domain Models, Artificial Intelligence (2017).

[38] A. Albore, H. Palacios, H. Geffner, A Translation-Based Approach to Contingent Planning., in: IJCAI, 2009, pp. 1623–1628.

[39] B. Bonet, H. Geffner, An Algorithm Better Than AO*?, in: AAAI, 2005, pp. 1343–1348.

[40] N. J. Nilsson, Principles of Artificial Intelligence, Morgan Kaufmann, 1980.

[41] M. Zakershahrak, Z. Gong, N. Sadassivam, Y. Zhang, Online Explanation Generation for Human-Robot Teaming, ICAPS Workshop on Explainable AI Planning (XAIP) (2019).

[42] N. J. Cooke, J. C. Gorman, C. W. Myers, J. L. Duran, Interactive Team Cognition, Cognitive Science (2013).

[43] M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?: Explaining the predictions of any classifier, in: KDD, 2016.

[44] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-Precision Model-Agnostic Explanations, in: AAAI, 2018.

[45] S. Sreedharan, S. Srivastava, S. Kambhampati, Hierarchical Expertise Level Modeling for User Specific Contrastive Explanations, in: IJCAI, 2018.

[46] S. Srivastava, S. J. Russell, A. Pinto, Metaphysics of Planning Domain Descriptions, in: AAAI, 2016.

[47] B. Marthi, S. J. Russell, J. A. Wolfe, Angelic Semantics for High-Level Actions, in: ICAPS, 2007, pp. 232–239.

[48] S. Vasileiou, W. Yeoh, T. C. Son, A Preliminary Logic-based Approach for Explanation Generation, in: ICAPS Workshop on Explainable AI Planning (XAIP), 2019.

[49] S. L. Vasileiou, A. Previti, W. Yeoh, On exploiting hitting sets for model reconciliation, AAAI, 2020.

[50] I. Lage, D. Lifschitz, F. Doshi-Velez, O. Amir, Exploring Computational User Models for Agent Policy Summarization, in: IJCAI, 2019.

[51] S. Sreedharan, A. O. Hernandez, A. P. Mishra, S. Kambhampati, Model-Free Model Reconciliation, in: IJCAI, 2019.

[52] S. Sreedharan, S. Srivastava, D. Smith, S. Kambhampati, Why Can't You Do That HAL? Explaining Unsolvability of Planning Tasks, in: IJCAI, 2019.

[53] M. Helmert, The Fast Downward Planning System, JAIR (2006).

[54] Y. Alkhazraji, M. Frorath, M. Grützner, T. Liebetraut, M. Ortlieb, J. Seipp, T. Springenberg, P. Stahl, J. Wülfing, M. Helmert, R. Mattmüller, Pyperplan, https://bitbucket.org/malte/pyperplan, 2016.

[55] International Planning Competition, IPC Competition Domains, `https://goo.gl/i35bxc`, 2011.

[56] T. Chakraborti, S. Sreedharan, S. Kambhampati, Balancing Explicability and Explanation in Human-Aware Planning, in: IJCAI, 2019.

[57] K. Allan, Common ground – aka "common knowledge", "mutual knowledge*", "shared knowledge", "assumed familiarity", "presumed background information", Handbook of Pragmatics (2013).

[58] R. Eifler, M. Cashmore, J. Hoffmann, D. Magazzeni, M. Steinmetz, A New Approach to Plan-Space Explanation: Analyzing Plan-Property Dependencies in Oversubscription Planning, AAAI, 2020.

[59] J. Hoffmann, D. Magazzeni, Explainable AI Planning (XAIP): Overview and the Case of Contrastive Explanation, in: Reasoning Web. Explainable Artificial Intelligence, 2019. Extended Abstract.

[60] T. Miller, Contrastive Explanation: A Structural-Model Approach, arXiv:1811.03163 (2018).

[61] T. Lombrozo, The Structure and Function of Explanations, Trends in Cognitive Sciences (2006).

[62] T. Lombrozo, Explanation and Abductive Inference, Oxford Handbook of Thinking and Reasoning (2012).

[63] M. Göbelbecker, T. Keller, P. Eyerich, M. Brenner, B. Nebel, Coming up With Good Excuses: What to do When no Plan Can be Found (2010).

[64] A. Herzig, V. Menezes, L. N. de Barros, R. Wassermann, On the Revision of Planning Tasks, in: ECAI, 2014.

[65] T. Eiter, E. Erdem, M. Fink, J. Senko, Updating Action Domain Descriptions, Artificial Intelligence Journal (2010).

[66] J. Porteous, A. Lindsay, J. Read, M. Truran, M. Cavazza, Automated Extension of Narrative Planning Domains with Antonymic Operators, in: AAMAS, 2015.

[67] M. Shvo, T. Q. Klassen, S. Sohrabi, S. A. McIlraith, Epistemic Plan Recognition, in: AAMAS, 2020.

[68] S. Sreedharan, T. Chakraborti, C. Muise, S. Kambhampati, Expectation-aware planning: A unifying framework for synthesizing and executing self-explaining plans for human-aware planning, AAAI, 2020.

[69] T. Miller, Social Planning – Reasoning with and about others, 2017. Invited Talk at IJCAI Workshop on Impedance Matching in Cognitive Partnerships.

[70] C. J. Muise, V. Belle, P. Felli, S. A. McIlraith, T. Miller, A. R. Pearce, L. Sonenberg, Planning Over Multi-Agent Epistemic States: A Classical Planning Approach., in: AAAI, 2015.

[71] C. Myers, J. Ball, N. Cooke, M. Freiman, M. Caisse, S. Rodgers, M. Demir, N. McNeese, Autonomous Intelligent Agents for Team Training, IEEE Intelligent Systems (2018).

[72] N. McNeese, M. Demir, E. Chiou, N. Cooke, G. Yanikian, Understanding the Role of Trust in Human-Autonomy Teaming, in: Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.

[73] T. Chakraborti, K. P. Fadnis, K. Talamadupula, M. Dholakia, B. Srivastava, J. O. Kephart, R. K. E. Bellamy, Planning and Visualization for a Smart Meeting Room Assistant – A Case Study in the Cognitive Environments Laboratory at IBM T.J. Watson Research Center, Yorktown, AI Communication (2019).

[74] S. Sreedharan, T. Chakraborti, C. Muise, Y. Khazaeni, S. Kambhampati, D3WA+: A Case Study of XAIP in a Model Acquisition Task, in: ICAPS, 2020.

[75] S. Grover, T. Chakraborti, S. Kambhampati, What Can Automated Planning do for Intelligent Tutoring Systems?, in: Scheduling and Planning Applications Workshop (SPARK) at ICAPS, 2018.

[76] J. Wetzel, K. VanLehn, D. Butler, P. Chaudhari, A. Desai, J. Feng, S. Grover, R. Joiner, M. Kong-Sivert, V. Patade, et al., The Design and Development of the Dragoon Intelligent Tutoring System for Model Construction: Lessons Learned, Interactive Learning Environments (2017).

[77] S. Kambhampati, T. Chakraborti, S. Sreedharan, A. Kulkarni, Synthesizing Explainable and Deceptive Behavior for Human-AI Interaction, in: AAAI Tutorial, 2020. https://yochan-lab.github.io/tutorial/AAAI-2020/.

[78] P. Madumal, T. Miller, L. Sonenberg, F. Vetere, Explainable Reinforcement Learning Through a Causal Lens, in: AAAI, 2020.

[79] S. Nikolaidis, P. Lasota, R. Ramakrishnan, J. Shah, Improved Human-Robot Team Performance through Cross-Training, an Approach Inspired by Human Team Training Practices, International Journal of Robotics Research (2015).

[80] D. Hadfield-Menell, S. J. Russell, P. Abbeel, A. Dragan, Cooperative Inverse Reinforcement Learning, in: NIPS, 2016.

[81] S. L. Vasileiou, W. Yeoh, Conditional updates of answer set programming and its application in explainable planning, in: AAMAS, 2020. Extended Abstract.

[82] M. Shvo, S. A. McIlraith, Towards Empathetic Planning, arXiv:1906.06436 (2019).

[83] V. Perera, S. P. Selvaraj, S. Rosenthal, M. Veloso, Dynamic Generation and Refinement of Robot Verbalization, in: RO-MAN, 2016.

[84] T. Chakraborti, S. Sreedharan, A. Kulkarni, S. Kambhampati, Projection-Aware Task Planning and Execution for Human-in-the-Loop Operation of Robots in a Mixed-Reality Workspace, in: IROS, 2018.

[85] R. G. Freedman, T. Chakraborti, K. Talamadupula, D. Magazzeni and J. D. Frank, User Interfaces and Scheduling and Planning: Workshop Summary and Proposed Challenges, in: AAAI 2018 Spring Symposium on Designing the User Experience of Artificial Intelligence, 2018.