Explanation Generation as Model Reconciliation in Multi-Model Planning

Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang and Subbarao Kambhampati
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
{tchakra2, ssreedh3, yzhan442, rao@asu.edu}

Abstract
When AI systems interact with humans in the loop, they are often called on to provide explanations for their plans and behavior. 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 the humans have domain and task models that differ significantly 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 human’s model, so as to make its plan be optimal with respect to that changed human model. We will study the properties of such explanations, present algorithms for automatically computing them, and evaluate the performance of the algorithms.

Introduction
There has been significant renewed interest recently in developing AI systems that can automatically provide explanations to humans in the loop. While much of the interest has been focused on learning systems that can explain their classification decisions, a related broader problem involves providing explanations in the context of human-AI interaction and human-in-the-loop decision making systems. In such scenarios, the automated agents are called upon to provide explanation of their behavior or plans [Langley, 2016].

Although explanation of plans has been investigated in the past (c.f. [Kambhampati, 1990; Sohrabi et al., 2011]), much of that work 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 required when the human is an expert “debugger” and is intimately familiar with the agent’s inards, it is completely unrealistic in most human-AI interaction scenarios, where the humans may have a domain and task model that differs significantly from that used by the AI system. This is illustrated in Figure 1 where the plans generated by the AI system with respect to its model need to be interpreted by the human with respect to his model. Of course, the AI system can avoid the need to provide explanations by being “explicable” [Zhang et al., 2017; Kulkarni et al., 2016] - i.e., generate plans that also make sense with respect to the humans’ model. Such explicability requirement however puts additional constraints on the agent’s plans, and may not always be feasible. When the robot’s plan is different from what the human would expect given his model of the world, the robot will be called on to “explain” its plan. We posit that such explanations should be seen as the robot’s attempt to move the human’s model to be in conformance with its own.

The primary contribution of this paper is to show how such model updates or explanations can be formulated concisely as the model reconciliation problem (MRP), which aims to make minimal changes to the human’s model to bring it closer to the robot’s model, in order to make the robot’s plan optimal with respect to this changed human’s model. One immediate complication in tackling MRP is that the human’s model is not directly made available to the robot, and will have to be learned instead (c.f. Zhang et al., 2017). The learned model may also be in a different form and at a different level of abstraction than the one used by the robot [Tian et al., 2016; Perera et al., 2016]. Nevertheless, for the purposes of this paper, we will assume that the human’s model is made available and is in PDDL format, just like the robot’s one. This allows us to focus on the explanation generation aspects.

In the rest of the paper, we will formalize the planning scenario depicted in Figure 1 as the multi-model planning set-
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).

ting, and characterize explanation generation as the model reconciliation problem (MRP) in this setting. We then present the notion of minimal explanations as the minimal set of updates needed to be made to the human model to make the robot’s plan optimal with respect to it. We present an A* search formulation for searching in the space of model updates to compute such minimal explanations. We also develop approximations to make this search more efficient. Finally, we move from the “one-shot” explanation problem to consider the issues of providing explanations during prolonged interactions between the human and the AI system. For this setting, we show that minimal explanation for one MRP may be rendered invalid given minimal explanations for a subsequent MRP. To avoid the loss of trust that such invalidated explanations can engender, we develop the stronger notion of minimally complete explanations. Finally, we present a preliminary evaluation of the efficiency of our algorithms for generating explanations in randomly generated problems in a few benchmark planning domains.

A Motivating Example Let us illustrate the concept of explanations via model reconciliation through an example based on the Fetch robot whose design requires it to tuck its arms and lower its torso or crouch before moving - which is not obvious to a human navigating it. This 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 in the following model snippet -

```pddl
(: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. Now, consider a problem with the following initial and goal states (here, identical for both the robot and the human) -

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

An optimal plan for the robot, in this case, involves a tuck action followed by a move -

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

The human, on the other hand, expects a much simpler model, as shown below. The move action does not have the preconditions for tucking the arm and lowering the torso, while tuck does not automatically lower the torso either.

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

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

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

Clearly, the original plan is no longer optimal (and hence explicable) here. One possible model update (i.e. explanation) that can mitigate this situation is -

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

This correction brings the human and the robot model closer, and is necessary and sufficient to make the robot’s plan optimal in the resultant domain. As explained before, we refer to such model corrections as multi-model explanations.

Related Work

Our view of explanation as a model reconciliation process is 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...” [Lombrozo, 2006]. This is achieved in our case by the appropriate change in the expectation of the model that is believed to have engendered the plan in question. Further, authors in [Lombrozo, 2012] also underline that explanations are “typically contrastive... the contrast provides a constraint on what should figure in a selected explanation...” - 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 [Sohrabi et al., 2011] where less costly plans are referred to as preferred explanations. The optimality criterion, however, makes the problem fundamentally different from model change algorithms in [Göbelbecker et al., 2010; Herzig et al., 2014; Perera et al., 2016; Bryce et al., 2016] which focus more on the feasibility of plans or correctness of domains. Finally, while the human-in-the-loop setting discussed here does bring back memories of mixed-initiative planners of the past [Ferguson et al., 1996; Ai-Chang et al., 2004], most of the work there involved the humans entering the land of planners; and not the other way around. Not surprisingly, it did not have the planner taking the human model into account in its planning or explanation.
The Multi-Model Planning (MMP) Setting

The Multi-Model Planning (MMP) Setting is given by the tuple \((M^R, M^H)\), where \(M^R = (D^R, I^R, G^R)\) is the planner’s model of the planning problem, while \(M^H = (H^R, I^H, G^H)\) is the human’s expectations of the same.

As we mentioned in the introduction, from the point of view of the planner, there can be two approaches to achieve common ground with the human in such settings - (1) **Change its own behavior in order to be explicable to the human** - in Zhang et al., 2017 [Kulkarni et al., 2016], the authors propose to modify the plan \(\pi\) itself so that \(C(\pi, M^H) \approx C^*_{M^H} \land \delta M(I^H, \pi) = G^H\). Thus the planner chooses to sacrifice optimality in order to make its behavior explicable to the human observer; and (2) **Bring the human’s model closer to its own by means of explanations in the form of model updates** - here, the planner does not change its own behavior, but rather corrects the human’s incorrect perception of its model via explanations. The task here is to find a modified planning problem \(\hat{M}\), closest to the human expectation \(M^H\), s.t. \(C(\pi, M^R) = C^*_{M^R}\), and also \(C(\pi, \hat{M}) = C^*_{\hat{M}}\). In this paper, we will only go into details of the explanation generation.
Algorithm 1 A* Search for Minimal Explanations

1: procedure ME-SEARCH
2: Input: MRP (π, (ℳᴴ, ℳᴴ))
3: Output: Explanation ℰ^{ME}
4: Procedure:
5: fringe ← Priority Queue()
6: c_list ← {∅} ▷ Closed list
7: π₀ᴴ ← π such that C(π, ℳᴴ) = C⁺₀ᴴ ▷ Optimal plan being explained
8: fringe.push(⟨ℳᴴ, {}, priority = 0⟩)
9: while True do
10: (ℳ, ℰ), c ← fringe.pop(ℳ)
11: if C(π₀ᴴ, ℳ) = C⁺₀ᴴ then return ℰ ▷ Return ℰ if π₀ᴴ optimal in ℳ
12: else for f ∈ Γ(ℳ) \ Γ(ℳᶠ) do ▷ Models that satisfy condition 1
13: λ ← (ℳ), {f(1)} ▷ Removes f from ℳ
14: if δ_uᴴ.M_H(Γ(ℳ), λ) ð c_list then
15: fringe.push(δ_uᴴ.M_H(Γ(ℳ), λ), ℰ \ λ, c + 1)
16: end for
17: for f ∈ Γ(ℳᶠ) \ Γ(ℳ) do ▷ Models that satisfy condition 2
18: λ ← (ℳ), {f(1)} ▷ Adds f to ℳ
19: if δ_uᴴ.M_H(Γ(ℳ), λ) ð c_list then
20: fringe.push(δ_uᴴ.M_H(Γ(ℳ), λ), ℰ \ λ, c + 1)
21: end for
22: end while
23: procedure Priority Queue.pop(ℳ)
24: candidates ← {((ℳ, ℰ), c) | c ∈ arg minₖ(⟨ℳ, ℰ⟩, c)}
25: pruned_list ← {} ▷ π_uᴴ ← π such that C(π, ℳ) = C⁺_uᴴ
26: for (⟨ℳ, ℰ⟩, c) ∈ candidates do
27: if ∃a ∈ π_uᴴ such that π⁻¹(Γ(ℳ) \ δ(Γ(ℳ))) ∈ {c_u} \ pre(a) \ eff⁺(a) \ eff⁻(a) then
28: π_uᴴ ← π \ U(1)(c_list)
29: pruned_list ← pruned_list \ ⟨⟨ℳ, ℰ⟩, c⟩ ▷ Candidates relevant to π₀ᴴ or π_uᴴ
30: else (⟨ℳ, ℰ⟩, c) ← U(1)(c_list)
31: end if
32: end for
33: end procedure

Model-space search for MEs

Our first attempt to solve for MEs involves A* search, similar to [Wayllace et al., 2016], in the space of models, as shown in Algorithm 1. Given an MRP, we start off with the initial state Γ(ℳᴴ) derived from the human’s expectation of a given planning problem ℳᴴ, and modify it incrementally until we arrive at a planning problem ℳ with C(π*, ℳ) = C⁺*₀ᴴ, 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 process. Also, we assign equal importance to all model changes. We can safely model differential importance of model updates by attaching costs to the edit actions - the algorithm remains unchanged.

We also employ a selection strategy of successor nodes to speed up search (by overloading the way the priority queue is being popped) by first processing model changes that are relevant to the actions in π₀ᴴ and π_uᴴ before the rest

Property 1 The successor selection strategy outlined in Algorithm 1 yields an admissible heuristic for model space search for minimal explanations.

Proof Let ℰ be the ME for an MRP problem and let ℰ⁺ be any intermediate explanation found by our search such that ℰ⁺ ⊆ ℰ, then the set ℰ \ ℰ⁺ must contain at least one λ related to actions in the set {a | a ∈ π₀ᴴ \ a ∈ π⁺} (where π⁺ is the optimal plan for the model ℳ where δ_uᴴ.M_H(Γ(ℳᴴ), ℰ⁺) = Γ(ℳ)). To see why this is true, consider a subset ℰ⁺ where ℰ⁺ = ℰ⁻ = |ℰ| - 1. If the action in ℰ \ ℰ⁺ does not belong to either π⁺ or π⁻ then it can not improve the cost of π⁺ in comparison to π⁻ and hence ℰ can not be the ME. Similarly we can show that this relation will hold for any size of ℰ⁺. We can leverage this knowledge about ℰ \ ℰ⁺ to create an admissible heuristic that will only consider the relevant changes at any given point of time (by giving very large values to all other changes).

We also note that the optimality criterion is relevant to both the cases where the human expectation is better, or when it is 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 (than expected optimal) 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 in the new model. Hence we emphasize the following property -

Property 2 If C(π⁺, ℳＨ) < minₖ C(π, ℳᴴ), then ensuring feasibility of the plan in the modified planning problem, i.e. δ_uᴴ(ℳ, π⁺) = ℓ, is a necessary but not a sufficient condition for ℳ = {D, ℰ, ℰ⁺} to yield a valid explanation.

Approximate Solution for MEs

Note that the biggest bottleneck in any of the algorithm discussed before is the check for optimality of a plan given a new model hypothesis. Thus the ability to check for necessary or sufficient conditions for optimality, without actually computing optimal plans in every node, can be used as a powerful tool to further prune the search tree. There has been some work in this regard in the context of plan monitoring [Fritz and McIlraith, 2007], which can prove to be an important asset to this method of search in future.

In the following section, we investigate an approximation that employs a few simple proxies to the optimality test. Specifically, we replace the equality test in line 12 of Algorithm 1 by the following rules -

1. δ_uᴴ(ℳ, π_uᴴ) = ℓ; and
2. C(π_uᴴ, ℳ) < C(π_uᴴ, ℳᴴ) or δ_uᴴ(ℳ, π_uᴴ) = ℓ; and
3. Each action contributes at least one causal link to π_uᴴ.

The first criterion simply ensures that the plan π_uᴴ originally computed by the planner is actually valid in the new hypothesis model. Criterion (2) requires that this plan has either become better in the new model or at least that the human’s expected plan π_uᴴ has been disproved. Finally, in Criterion (3), we ensure that for each action aᵢ ∈ π_uᴴ there exists an effect p that satisfies the precondition of at least one action aⱼ (where aᵢ ⊆ aⱼ ⊆ a_k) such that p ∈ eff⁻(aⱼ).
Algorithm 2 Search for Minimally Complete Explanations

1: procedure MCE-SEARCH
2: Input: MRP $(\pi^*, (\mathcal{M}^R, \mathcal{M}^H))$
3: Output: Explanation $\mathcal{E}^{\text{MCE}}$
4: Procedure:
5: $\mathcal{E}^{\text{MCE}} \leftarrow \{\}$
6: fringe $\leftarrow \text{PriorityQueue}\{\}$ → Closed list
7: $\lambda, \pi \leftarrow \{\}$ → List of incorrect model changes
8: fringe.push($\mathcal{M}^R, \{\}$, priority = 0)
9: while fringe is not empty do
10: $(\tilde{\mathcal{M}}, \mathcal{E}), \pi \leftarrow \text{fringe.pop}(\tilde{\mathcal{M})}$
11: if $C(\pi^*, \tilde{\mathcal{M}}) > C^*_\mathcal{M}$ then
12: $\lambda \leftarrow \lambda \cup (\Gamma(\tilde{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^R))$ → Updating $\lambda$
13: $\pi \leftarrow \{\pi, \lambda\}$ → Updating $\pi$
14: for $f \in \Gamma(\tilde{\mathcal{M}}) \setminus \Gamma(\mathcal{M}^H)$ do → Models that satisfy condition 1
15: $\lambda \leftarrow (\lambda, \{f\})$ → Removes $f$ from $\lambda$
16: $h, \pi \leftarrow h, \pi \cup (\Gamma(\tilde{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^R))$ → Adding $h$ to $\pi$
17: $\lambda \leftarrow (\lambda, \{f\})$ → Removes $f$ from $\lambda$
18: if $\delta_{\mathcal{M}^R, \mathcal{M}^H}(\Gamma(\tilde{\mathcal{M}}), \lambda) \notin \varepsilon_{\lambda}$
19: $\mathcal{E}^{\text{MCE}} \leftarrow \max_{\lambda} (\{\mathcal{E}^{\text{MCE}}, \mathcal{E})$
20: for $f \in \Gamma(\mathcal{M}^H) \setminus \Gamma(\mathcal{M}^R)$ do → Models that satisfy condition 2
21: $\lambda \leftarrow (\lambda, \{f\})$ → Adds $f$ to $\lambda$
22: $\lambda \leftarrow (\lambda, \{f\})$ → Removes $f$ from $\lambda$
23: if $\delta_{\mathcal{M}^R, \mathcal{M}^H}(\Gamma(\tilde{\mathcal{M}}), \lambda) \notin \varepsilon_{\lambda}$
24: $\mathcal{E}^{\text{MCE}} \leftarrow \max_{\lambda} (\{\mathcal{E}^{\text{MCE}}, \mathcal{E})$
25: end while

Figure 3: Illustration of model space search for ME & MCE.

get away with minimal explanations, since these are typically easier to compute (see Evaluations). However, if the human does look back at the validity of previous explanations, then MCEs become necessary since inconsistency on the part of the robot’s can lead to quick erosion of trust and credibility.

A minimally complete explanation (MCE) is the shortest explanation $\mathcal{E}^{\text{MCE}} = \arg \min_{\mathcal{E}} |\Gamma(\mathcal{M})| \Delta |\Gamma(\mathcal{M}^H)|$, given the MRP $(\pi^*, (\mathcal{M}^R, \mathcal{M}^H))$, where $\exists \mathcal{M}$ s.t. $\Gamma(\mathcal{M}) \Delta \Gamma(\mathcal{M}^H) \subset \Gamma(\mathcal{M}) \Delta \Gamma(\mathcal{M}^H)$ with $C(\pi^*, \mathcal{M}) > C^*_\mathcal{M}$.

This means that beyond the model obtained from the minimally complete 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 the planner’s planning problem $\mathcal{M}^R$ and still find a model $\tilde{\mathcal{M}}$ where $C(\pi^*, \tilde{\mathcal{M}}) = C^*_\mathcal{M}$ - we are going to use this property in the search for MCEs.

Property 5 $\mathcal{E}^{\text{MCE}} = \arg \max_{\mathcal{E}} |\Gamma(\mathcal{M})| \Delta |\Gamma(\mathcal{M}^R)|$ such that $C(\pi^*, \mathcal{M}) = C^*_\mathcal{M}$ and $\forall \mathcal{M} \Gamma(\mathcal{M}) \Delta \Gamma(\mathcal{M}^R) \subset \Gamma(\mathcal{M}) \Delta \Gamma(\mathcal{M}^R)$.

Model-space search for MCEs: To compute MCEs, we take a similar approach to the model-space search for MEs described before, but this time starting from the planner’s planning problem $\mathcal{M}$. 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). The search process, described in detail in Algorithm 2, requires a search over the entire model space. We can leverage Property 5 to reduce our search space. Starting from $\mathcal{M}^R$, given a set of model changes $\mathcal{E}$ where $\delta_{\mathcal{M}^R, \mathcal{M}^H}(\Gamma(\mathcal{M}^R), \mathcal{E}) = \Gamma(\mathcal{M})$ and $C(\pi^*, \mathcal{M}) > C^*_\mathcal{M}$, we update $\mathcal{E}^{\text{MCE}}$ with $\mathcal{E}$. The resulting MCE explanation is all the possible model changes that do not appear in the set $\mathcal{E}^{\text{MCE}}$.

Note that, unlike in the case of MEs, since MCEs guarantee that further explanations are always consistent with the current plan being explained, it does not make sense to approximate MCEs. Figure 3 contrasts ME with MCE search. ME search starts from $\mathcal{M}^R$, computes updates $\tilde{\mathcal{M}}$ towards $\mathcal{M}^R$ and returns the first node (indicated in orange) where

Consistency of Explanations

Note that a minimal explanation for an MRP can be rendered invalid given the minimal explanations for a subsequent MRP. 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 (tucked away), 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. Thus -

Property 4 MEs are non-monotonic, i.e. it is possible to have $\Gamma(\tilde{\mathcal{M}}) \Delta \Gamma(\mathcal{M}^H) \supset \Gamma(\mathcal{M}) \Delta \Gamma(\mathcal{M}^H)$ such that $C(\pi^*, \mathcal{M}) = C^*_\mathcal{M}$ but $C(\pi^*, \mathcal{M}) \neq C^*_\mathcal{M}$.

Consider the following explanation instead -

Explanation >>
TUCK-has-add-effect-CROUCHED
MOVE-LONG-TASK-has-precondition-CROUCHED

This new explanation ensures that the robot’s plan remains optimal for this problem, irrespective of any future changes to the model, and is the smallest possible among all such explanations. The question of consistency brings us to the notion of minimally complete explanations. Note that, in cases where only one shot explanations are required, the robot can

$\delta_{\mathcal{M}^R}(\phi, \pi^*_R) = |\mathcal{G}|$. Now let us consider a cheaper plan $\pi^*_R = (a_0, a_1, \ldots, a_{n-1}, a_n, a_{n+1})$. Since $a_i$ does not contribute any causal links to the original plan $\pi^*_R$, we will also have $\delta_{\mathcal{M}^R}(\phi, \pi^*_R) = |\mathcal{G}|$. This contradicts our original assumption of $\pi^*_R$ being optimal, hence proved.

δ_M(φ, π_M) = G. Now let us consider a cheaper plan π_R = (a_0, a_1, \ldots, a_{n-1}, a_n, a_{n+1}). Since a_i does not contribute any causal links to the original plan π_R, we will also have δ_M(φ, π_R) = G. This contradicts our original assumption of π_R being optimal, hence proved.

δ_M(φ, π_M) = G. Now let us consider a cheaper plan π_R = (a_0, a_1, \ldots, a_{n-1}, a_n, a_{n+1}). Since a_i does not contribute any causal links to the original plan π_R, we will also have δ_M(φ, π_R) = G. This contradicts our original assumption of π_R being optimal, hence proved.

δ_M(φ, π_M) = G. Now let us consider a cheaper plan π_R = (a_0, a_1, \ldots, a_{n-1}, a_n, a_{n+1}). Since a_i does not contribute any causal links to the original plan π_R, we will also have δ_M(φ, π_R) = G. This contradicts our original assumption of π_R being optimal, hence proved.
$C(\pi^*, \hat{M}) = C^{\pi^*}_{\hat{M}}$, MCE search starts from $\mathcal{M}^R$ and moves towards $\mathcal{M}^H$. It finds the longest path (indicated in blue) where $C(\pi^*, \hat{M}) = C^{\pi^*}_{\hat{M}}$ for all $\hat{M}$ in the path. The MCE (shown in green) is the rest of the path towards $\mathcal{M}^H$.

**Empirical Evaluations**

Our explanation generation system (as previewed in the Fetch domain) integrates calls to the Fast-Downward planner [Helmert, 2006] for planning support at the search nodes, and VAL [Howey et al., 2004] for plan validation support, and pyperplan [Alkhazraji et al., 2016] for parsing PDDL domains. The results reported here are from experiments run on a 12 core Intel(R) Xeon(R) CPU with an E5-2643 v3@3.40GHz processor and a 64G RAM. The code will be released after the double-blind review period.

We use three planning domains - 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. Though the following experiments are only pertaining 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, was done in the lifted representation of the domain.

**Table 1.** In Table 1, we make changes at random to the domains and measure the number of explanations produced and the time taken (in secs) to produce them, against the ground truth. Observe the gains produced by the heuristic in terms of time spent on each problem. Further, note how close the approximate version of MEs are to the exact solutions. As expected, MCE search is significantly costlier to compute than ME. However, note that both MEs and MCEs 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 minimal explanations as opposed to dumping the entire model difference on the human. A general rule of thumb is -

$$\text{approx. } ME \leq |\text{exact } ME| < |\text{MCE}| \ll |\mathcal{M}^R \Delta \mathcal{M}^H|$$

It is interesting to note that time required to calculate MCE in the Logistics problems is much lower than the time required to calculate the corresponding ME. This is because for most of these problems, even a single change in the robot model made the robot plan be no longer optimal. This meant that the search ended after checking all possible unit changes. In general, closer ME is to the total number of changes shorter the MCE search would be.

**Table 2.** We now make progressively higher number of changes in the human model of the BlocksWorld domain, and illustrate the relative time (in secs) taken to search for exact MEs in 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 finding useful approximations to the explanation generation problem.

**Table 3.** Finally, we demonstrate how Property 5 reduces the number of nodes that need to be searched to find MCEs 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 $|\mathcal{P}(\mathcal{M}^R \Delta \mathcal{M}^H)|$.

**Conclusions and Future Work**

In this paper, we argued that to explain its plans to the human agents in the loop, an AI system needs to explicitly acknowledge that the human may be using a different model than it does. Explanations in this multi-model setting become a process of identifying and reconciling the relevant differences between the models. We showed that minimal explanations capture this adequately for one-shot explanations, while minimally complete explanations provide consistency for continual interactions between the human and AI system. One immediate future direction is to allow for human’s models that are of different form and/or level of abstraction than the robot’s, so as to allow effective learning of the human’s models (c.f. Tian et al., 2016; Zhang et al., 2016). In cases where the ground truth is not known, the explanation process might also need to consider distributions over relevant models, and iterative refinement of the same via dialog with the human. Also note that we insisted that explanations must be consistent with the planner’s model. If this requirement is relaxed, it allows the planner to generate alternative explanations that it knows are not true, and thus deceive the human. 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 [Boella et al., 2009] has received very little attention [van Ditmarsch, 2014] and affords a rich set of exciting research problems.
References

[Ai-Chang et al., 2004] Mitchell Ai-Chang, John Bresina, Len Charest, Adam Chase, JC-J Hsu, Åri Jonsson, Bob Kaneisky, Paul Morris, Kanna Rajan, Jeffrey Yglesias, et al. Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems*, 19(1):8–12, 2004.

[Alkhazraji et al., 2016] Yusra Alkhazraji, Matthias Frorath, Markus Grützner, Thomas Liebetraut, Manuela Ortlieb, Jendrik Seipp, Tobias Springeren, Philip Stahl, Jan Wülfing, Malte Helmert, and Robert Mattmüller. Pyper-plan. https://bitbucket.org/malte/pyperplan, 2016.

[Boella et al., 2009] Guido Boella, Jan Broersen, Leendert van der Torre, and Serena Villata. *Representing Excuses in Social Dependence Networks*, pages 365–374. Springer Berlin Heidelberg, 2009.

[Bryce et al., 2016] Dan Bryce, J. Benton, and Michael W. Boldt. Maintaining evolving domain models. In *IJCAI*, 2016.

[Ferguson et al., 1996] George Ferguson, James F Allen, Bradford W Miller, et al. Train:95. Towards a mixed-initiative planning assistant. In *AIPS*, pages 70–77, 1996.

[Fritz and McIlraith, 2007] Christian Fritz and Sheila A McIlraith. Monitoring plan optimality during execution. In *ICAPS*, pages 144–151, 2007.

[Göbelbecker et al., 2010] Moritz Göbelbecker, Thomas Keller, Patrick Eyerich, Michael Brenner, and Bernhard Nebel. Coming up with good excuses: What to do when no plan can be found. 2010.

[Helmert, 2006] Malte Helmert. The fast downward planning system. *J. Artif. Intell. Res.(JAIR)*, 26:191–246, 2006.

[Herzig et al., 2014] Andreas Herzig, Viviane Menezes, Leliane Nunes de Barros, and Renata Wassermann. On the revision of planning tasks. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence, ECAI’14*, 2014.

[Howey et al., 2004] Richard Howey, Derek Long, and Maria Fox. Val: Automatic plan validation, continuous effects and mixed initiative planning using pddl. In *Tools with Artificial Intelligence, 2004, ICTAI 2004. 16th IEEE International Conference on*, pages 294–301. IEEE, 2004.

[Kambhampati, 1990] Subbarao Kambhampati. A classification of plan modification strategies based on coverage and information requirements. In *AAAI 1990 Spring Symposium on Case Based Reasoning*, 1990.

[Kulkarni et al., 2016] Anagha Kulkarni, Tathagata Chakraborti, Yantian Zha, Satya Gautam Vadlamudi, Yu Zhang, and Subbarao Kambhampati. Explicable robot planning as minimizing distance from expected behavior. *CoRR*, abs/1611.05497, 2016.

[Langley, 2016] Pat Langley. Explainable agency in human-robot interaction. In *AAAI Fall Symposium Series*, 2016.

[Lombrozo, 2006] Tania Lombrozo. The structure and function of explanations. *Trends in Cognitive Sciences*, 10(10):464 – 470, 2006.

[Lombrozo, 2012] Tania Lombrozo. Explanation and abductive inference. *Oxford handbook of thinking and reasoning*, pages 260–276, 2012.

[McDermott et al., 1998] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl-the planning domain definition language. 1998.

[Perera et al., 2016] Vittorio Perera, Sai P Selveraj, Stephanie Rosenthal, and Manuela Veloso. Dynamic generation and refinement of robot verbalization. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 212–218. IEEE, 2016.

[Riabov et al., 2014] Anton Riabov, Shirin Sohrabi, and Octavian Udrea. New algorithms for the top-k planning problem. In *Proceedings of the Scheduling and Planning Applications woRKshop (SPARK) at the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 10–16, 2014.

[Russell et al., 2003] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River, 2003.

[Sohrabi et al., 2011] Shirin Sohrabi, Jorge A. Baier, and Sheila A. McIlraith. Preferred explanations: Theory and generation via planning. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11)*, pages 261–267, San Francisco, USA, August 2011.

[Tian et al., 2016] Xin Tian, Hankz Hankui Zhuo, and Subbarao Kambhampati. Discovering underlying plans based on distributed representations of actions. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1135–1143. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[van Ditmarsch, 2014] Hans van Ditmarsch. *The Ditmarsch Tale of Wonders*, pages 1–12. Springer International Publishing, Cham, 2014.

[Wayllace et al., 2016] Christabel Wayllace, Ping Hou, William Yeoh, and Tran Cao Son. Goal recognition design with stochastic agent action outcomes. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3279–3285, 2016.

[Zhang et al., 2016] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explicability for robot task planning. In *Proceedings of the RSS Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics*, 2016.

[Zhang et al., 2017] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explicability and predictability for robot task planning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017.