Explanations as Model Reconciliation - A Multi-Agent Perspective

Sarath Sreedharan* and **Tathagata Chakraborti*** and **Subbarao Kambhampati**

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ 85281 USA

{ ssreedh3, tchakra2, rao } @ asu.edu

Abstract

In this paper, we demonstrate how a planner (or a robot as an embodiment of it) can explain its decisions to *multiple* agents in the loop together considering not only the model that it used to come up with its decisions but also the (often misaligned) models of the same task that the other agents might have had. To do this, we build on our previous work on *multi-model explanation generation* (Chakraborti et al. 2017b) and extend it to account for settings where there is uncertainty of the robot's model of the explainee and/or there are multiple explainees with different models to explain to. We will illustrate these concepts in a demonstration on a robot involved in a typical search and reconnaissance scenario with another human teammate and an external human supervisor.

In (Chakraborti et al. 2017b) we showed how a robot can explain its decisions to a human in the loop who might have a different understanding of the same problem (either in terms of the agent's knowledge or intentions, or in terms of its capabilities). These explanations are intended to bring the human's mental model closer to the robot's estimation of the ground truth – we refer to this as the *model reconciliation process* by the end of which a plan that is optimal in the robot's model is also optimal in the human's updated mental model. We also showed how this process can be achieved successfully while transferring the minimum number of model updates possible via what we call *minimally complete explanations* or MCEs. Such techniques can be essential contributors to the dynamics of trust and teamwork in human-agent collaborations by significantly lowering the communication overhead between agents while at the same time providing the right amount of information to keep the agents on the same page with respect to their understanding of each others' tasks and capabilities – thereby reducing the cognitive burden on the human teammates and increasing their situational awareness.

The process of model reconciliation is illustrated in Figure 1. The robot's model, which is its ground truth, is represented by $\mathcal{M}^R$ (note: "model" of a planning problem includes the state and goals information as well as the domain or action model) and $\pi^*_{\mathcal{M}^R}$ is the optimal plan in it. A human H who is interacting with it may have a different model $\mathcal{M}^R_h$

*Authors marked with asterix contributed equally.

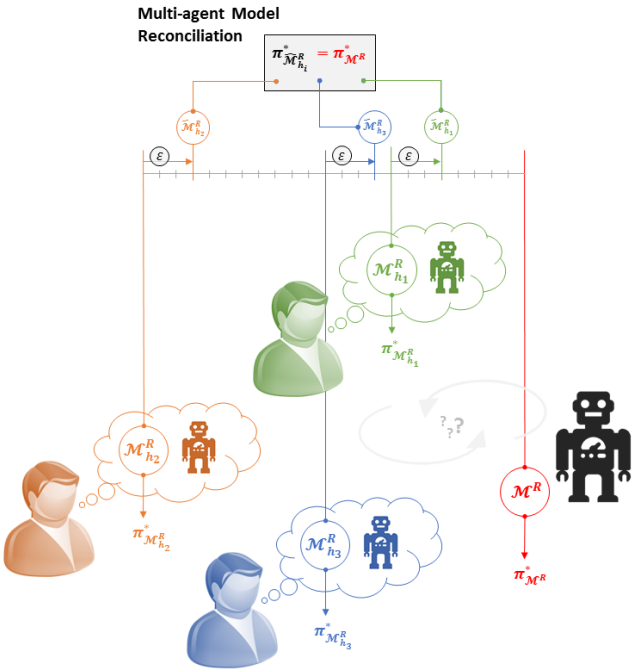

Figure 1: The model reconciliation process in case of model uncertainty or multiple explainees.

of the same planning problem, and the optimal plan $\pi^*_{\mathcal{M}^R_h}$ in the human's model can diverge from that of the robot's leading to the robot needing to explain it's decision to the human. As explained above, a multi-model explanation is an update or correction to the human's mental model to a new model $\widehat{\mathcal{M}}^R_h$ where the optimal plan $\pi^*_{\widehat{\mathcal{M}}^R_h}$ is equivalent to $\pi^*_{\mathcal{M}^R}$.

Imagine that the planner is now required to explain the same problem to multiple different human teammates H_i, or if the model of the human is not known with certainty (which is an equivalent setting with multiple possible models). The robot can, of course, call upon the previous service to compute MCEs for each such configuration. However, this can result in situations where the explanations computed for individual models independently are not consistent across the all the possible target domains. In the case of multiple teammates being explained to, this may cause confusion and loss

of trust; and in the case of model uncertainty, such an approach cannot even guarantee that the resulting explanation will be an acceptable explanation in the real domain. Instead, we want to find an explanation such that $\forall i\ \pi^*_{\widehat{\mathcal{M}}^R_{h_i}} \equiv \pi^*_{\mathcal{M}^R}$, i.e. a single model update that makes the given plan optimal 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 a optimal planning problem. However, it turns out that the exact same search strategy can be employed here as well by modifying the way in which the models are represented and the equivalence criterion is computed during the search process.

Thus, in this paper, we (1) outline how uncertainty over models in the multi-model planning setting can be represented in the form of *annotated* models; (2) show how the search for a minimally complete explanation in the revised setting can be compiled to the original MCE search based on this representation; and (3) demonstrate these concepts on a typical search and reconnaissance setting involving a robot and its human teammate internal to a disaster scene and an external human commander supervising the proceedings.

Background

In this section, we provide a brief introduction to the classical planning problem and its evolution towards "model-lite" planning to handle model uncertainty.

A Classical Planning Problem is a tuple $\mathcal{M} = \langle \mathcal{D}, \mathcal{I}, \mathcal{G} \rangle$[1] 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. $\delta_{\mathcal{M}}(s, a) \models \bot$ *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. The cumulative transition function is given by $\delta_{\mathcal{M}}(s, \langle a_1, a_2, \ldots, a_n \rangle) = \delta_{\mathcal{M}}(\delta_{\mathcal{M}}(s, a_1), \langle a_2, \ldots, a_n \rangle)$.

This forms the classical definition of a planning problem (Russell and Norvig 2003) whose models are represented in the syntax of PDDL (McDermott et al. 1998). 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 $C^*_{\mathcal{M}}$.

In (Nguyen, Sreedharan, and Kambhampati 2017) the authors introduced an update to the standard representation of planning problems to an *annotated* model or PDDL to account for uncertainty over the definition of the planning model. In addition to the standard preconditions and effects associated with the definition of actions, this introduces the notion of *possible* preconditions and effects which may or may not be realized in practice. Such representations are relevant especially in the context of learning human mental models, where uncertainty after the learning process can be represented in terms of annotated models as in (Bryce, Benton, and Boldt 2016).

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

An *instantiation* of an annotated model ${}^a\mathcal{M}$ is a classical planning model where a subset of the possible conditions have been realized, and is thus given by the tuple $ins({}^a\mathcal{M}) = \langle \mathcal{D}, \mathcal{I}, \mathcal{G} \rangle$ 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 $A \ni a = \langle c_a, \mathit{pre}(a) \leftarrow \mathit{pre}(a) \cup \chi;\ \chi \subseteq \widetilde{\mathit{pre}}(a), \mathit{eff}^{\pm}(a) \leftarrow \mathit{eff}^{\pm}(a) \cup \chi;\ \chi \subseteq \widetilde{\mathit{eff}}^{\pm}(a) \rangle$. Given an annotated model with k possible conditions, there may be 2^k such instantiations, which forms its *completion set* (Nguyen, Sreedharan, and Kambhampati 2017).

The Multi-Model Planning Setting

The *multi-model* planning paradigm (Chakraborti et al. 2017b) 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. In such settings, when a planner's optimal plans diverge from human expectations[2], the planner can attempt corrections to the human's mental model to resolve the inoptimality by participating in what we call the *model reconciliation* process. Thus –

A Multi-Model Planning (MMP) Setting is the tuple $\Phi = \langle \mathcal{M}^R, \mathcal{M}^R_h \rangle$, where $\mathcal{M}^R = \langle D^R, \mathcal{I}^R, \mathcal{G}^R \rangle$ is the planner's model of a planning problem, while $\mathcal{M}^R_h = \langle D^R_h, \mathcal{I}^R_h, \mathcal{G}^R_h \rangle$ is the human's expectations of the same.

The Model Reconciliation Problem (MRP) is the tuple $\Psi = \langle \pi, \Phi \rangle$, given an MMP ϕ, where $C(\pi, \mathcal{M}^R) = C^*_{\mathcal{M}^R}$.

A solution to an MRP is the set of model changes $\mathcal{E}$ or a *multi-model explanation*, such that

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

(2) $C(\pi, \widehat{\mathcal{M}}^R_h) = C^*_{\widehat{\mathcal{M}}^R_h}$.

A Minimally Complete Explanation (MCE) is the shortest explanation that satisfies conditions (1) and (2).

[1] Note that the definition of a planning "model" includes the action model as well as the initial and goal states of an agent.

[2] This is modeled here in terms of cost optimality, but in general this can be any preference metric like plan or causal link similarity.

As we mentioned before, in the case of an model uncertainty / multiplicity, we want conditions (1) and (2) to hold for all instances of the model being explained to. In the following discussion, we are going to show how this can be achieved by a modified version of the original MCE-search in (Chakraborti et al. 2017b) using annotated models.

MRP for Model Uncertainty / Multiplicity

We represent the uncertainty or multiplicity of the model of the explainee in terms of the annotated model introduced in the previous section – by making preconditions and effects that appear in all possible models be necessary ones, and those that appear in just a subset to be possible ones. Let the set of models under consideration (one belonging to each explainee h_i) be $\{\mathcal{M}^R_{h_i}\}$. From this set of models we construct the following annotated model –

${}^a\mathcal{M}^R_H = \langle {}^a\mathcal{D}, {}^a\mathcal{I}, {}^a\mathcal{G}\rangle$ with domain ${}^a\mathcal{D} = \langle F, {}^aA\rangle$ and initial and goal states ${}^a\mathcal{I} = \langle \mathcal{I}^0, \mathcal{I}^+\rangle$, ${}^a\mathcal{G} = \langle \mathcal{G}^0, \mathcal{G}^+\rangle$ where

- Action ${}^aA \ni a = \langle c_a, \textit{pre}(a), \widetilde{\textit{pre}}(a), \textit{eff}^{\pm}(a), \widetilde{\textit{eff}}^{\pm}(a)\rangle$ where c_a is the action cost[3] and –
  - $\textit{pre}(a) = \{f \mid \forall i\ f \in \textit{pre}(a_i)\}$
  - $\widetilde{\textit{pre}}(a) = \{f \mid \exists i\ f \notin \textit{pre}(a_i)\ \wedge \exists i\ f \in \textit{pre}(a_i)\}$
  - $\textit{eff}^{\pm}(a) = \{f \mid \forall i\ f \in \textit{eff}^{\pm}(a_i)\}$
  - $\widetilde{\textit{eff}}^{\pm}(a) = \{f \mid \exists i\ f \notin \textit{eff}^{\pm}(a_i)\ \wedge \exists i\ f \in \textit{eff}^{\pm}(a_i)\}$
- $\mathcal{I}^0 = \{f \mid \forall i\ f \in \mathcal{I} \in \mathcal{M}^R_{h_i}\}$
- $\mathcal{I}^+ = \{f \mid \exists j\ f \notin \mathcal{I}_j \wedge \exists i\ f \in \mathcal{I}_i;\ \mathcal{I}_i \in \mathcal{M}^R_{h_i},\ \mathcal{I}_j \in \mathcal{M}^R_{h_j}\}$
- $\mathcal{G}^0 = \{f \mid \forall i\ f \in \mathcal{G} \in \mathcal{M}^R_{h_i}\}$
- $\mathcal{G}^+ = \{f \mid \exists j\ f \notin \mathcal{G}_j \wedge \exists i\ f \in \mathcal{G}_i;\ \mathcal{G}_i \in \mathcal{M}^R_{h_i},\ \mathcal{G}_j \in \mathcal{M}^R_{h_j}\}$

Alternatively, consider ${}^a\mathcal{M}^R_H$ as the culmination of a model learning process and the model set $\{\mathcal{M}^R_{h_i}\}$ is the completion set of ${}^a\mathcal{M}^R_H$. As mentioned earlier, we intend to find a single explanation that is a satisfactory explanation for the entire set of models, without having to iterate the standard MRP process over all possible models while coming up with an explanation that can satisfy all of them.

$\mathcal{M}_{max}$ & $\mathcal{M}_{min}$ Models

We begin by defining two models – the most relaxed model $\mathcal{M}_{max}$ possible 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 through out 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, produces 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 exist, and we intent to find the smallest one. ${}^a\mathcal{M}^R_H$ 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 ${}^a\mathcal{I}$
- goal state $\mathcal{G} \leftarrow \mathcal{G}^0$; given ${}^a\mathcal{G}$
- $\forall a \in A$
  - $\textit{pre}(a) \leftarrow \textit{pre}(a);\ a \in {}^aA$
  - $\textit{eff}^+(a) \leftarrow \textit{eff}^+(a) \cup \widetilde{\textit{eff}}^+(a);\ a \in {}^aA$
  - $\textit{eff}^-(a) \leftarrow \textit{eff}^-(a);\ a \in {}^aA$

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

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

As explained before, $\mathcal{M}_{max}$ is a model where all the positive conditions hold and it is easiest to achieve the goal, and vice versa for $\mathcal{M}_{min}$. Note that these definitions might end up creating inconsistencies in the models (e.g. in an annotated model for the `BlocksWorld` domain, the definition of `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 1 For a given MRP $\Psi = \langle \pi, \langle \mathcal{M}^R, \{\mathcal{M}^R_{h_i}\}\rangle\rangle$, if the plan π is optimal in $\mathcal{M}_{max}$ and executable in $\mathcal{M}_{min}$, then conditions (1) and (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.

Model-Space Search

We will employ use a modified version of the *model space A* search* in (Chakraborti et al. 2017b) to calculate the minimal explanation in the presence of model uncertainty / multiplicity. We define the following state representation, as outline in (Chakraborti et al. 2017b), over planning problems for our model-space search algorithm –

$$\mathcal{F} = \{\textit{init-has-f} \mid \forall f \in F^R_h \cup F^R\} \cup \{\mathcal{G}\textit{-has-f} \mid \forall f \in F^R_h \cup F^R\}$$
$$\bigcup_{a \in A^R_h \cup A^R} \{\textit{a-has-precondition-f}, \textit{a-has-add-effect-f}, \textit{a-has-del-effect-f} \mid \forall f \in F^R_h \cup F^R\}$$
$$\cup \{\textit{a-has-cost-}c_a \mid a \in A^R_h\} \cup \{\textit{a-has-cost-}c_a \mid a \in A^R\}.$$

[3] Note that for the time being we ignore uncertainty over cost of an action. Refer to (Nguyen et al. 2012) for a possible way to address this by computing *diverse* plans.

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 -

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

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

We 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 $\lambda : \mathcal{F} \rightarrow \mathcal{F}$ such that $|s_1 \Delta s_2| = 1$, where the new transition or edit function is given by $\delta_{\mathcal{M}_1,\mathcal{M}_2}(s_1, \lambda) = s_2$ such that `condition 1` : $s_2 \setminus s_1 \subseteq \Gamma(\mathcal{M}_2)$ and `condition 2` : $s_1 \setminus s_2 \nsubseteq \Gamma(\mathcal{M}_2)$ are satisfied. This means that model change actions can only make a single change to a domain at a time, and *all these changes are consistent with the model of the planner.* The solution to a model-space search problem is given by a set of edit functions $\{\lambda_i\}$ that can transform the model $\mathcal{M}_1$ to the model $\mathcal{M}_2$, i.e. $\delta_{\mathcal{M}_1,\mathcal{M}_2}(\Gamma(\mathcal{M}_1), \{\lambda_i\}) = \Gamma(\mathcal{M}_2)$. Thus, for a given MRP Ψ, an MCE is the smallest 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 $C(\pi, \widehat{\mathcal{M}}) = C^*_{\widehat{\mathcal{M}}}$, i.e. $\mathcal{E}^{MCE} = \arg\min_{\mathcal{E}} |\Gamma(\widehat{\mathcal{M}}) \Delta \Gamma(\mathcal{M}_h^R)|$.

Our MEGA Algorithm

The proposed search procedure is presented in Algorithm 1. The search closely follows the MCE search defined in (Chakraborti et al. 2017b) with minimal additions[4] to accommodate the annotated model. We start the search by first creating the corresponding $\mathcal{M}_{max}$ and $\mathcal{M}_{min}$ model for the given annotated model ${}^a\mathcal{M}_h^R$. 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 1, 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 difference between the robot model and $\mathcal{M}_{min}$ and the difference 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.

[4]Similar to the new MCE search, we can also adapt MME, approximate MCE and even the heuristic in (Chakraborti et al. 2017b) to work with annotated PDDL models with minimal changes.

Algorithm 1 MEGA

procedure MCE-SEARCH
Input: MRP $\langle \pi^*, \langle \mathcal{M}^R, {}^a\mathcal{M}_h^R \rangle \rangle$
Output: Explanation $\mathcal{E}^{MCE}$
Procedure:
  fringe $\leftarrow$ `Priority_Queue()`
  c_list $\leftarrow \{\}$ ▷ Closed list
  $\pi_R^* \leftarrow \pi^*$ ▷ Optimal plan being explained
  $\mathcal{M}_{max},\ \mathcal{M}_{min} \leftarrow ({}^a\mathcal{M}_h^R)$ ▷ Proposition 2
  fringe.push($\langle \mathcal{M}_{min}, \mathcal{M}_{max}, \{\}\rangle,$ 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 1
    else
      c_list $\leftarrow$ 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 $\delta_{\mathcal{M}^H,\mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max})\rangle, \lambda) \notin$ c_list **then**
          fringe.push($\langle \delta_{\mathcal{M}^H,\mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max})\rangle, \lambda), \mathcal{E} \cup \lambda\rangle,\ c+1$)
      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 $\delta_{\mathcal{M}^H,\mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max})\rangle, \lambda) \notin$ c_list **then**
          fringe.push($\langle \delta_{\mathcal{M}^H,\mathcal{M}^R}(\langle \Gamma(\widehat{\mathcal{M}}_{min}), \Gamma(\widehat{\mathcal{M}}_{max})\rangle, \lambda), \mathcal{E} \cup \lambda\rangle,\ c+1$)

Proposition 2 $\mathcal{M}_{max}$ and $\mathcal{M}_{min}$ only need to be computed once before the search – i.e. with a model update $\mathcal{E}$ to $\{\mathcal{M}_{h_i}^R\}$, $\mathcal{M}_{max} \leftarrow \mathcal{M}_{max} + \mathcal{E}$ and $\mathcal{M}_{min} \leftarrow \mathcal{M}_{min} + \mathcal{E}$ for the new model set.

Following Proposition 2, these models form the new $\mathcal{M}_{min}$ and $\mathcal{M}_{max}$ models for the set 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.

Demonstration

We will now demonstrate MEGA on a robot performing an Urban Search And Reconnaissance (USAR) task - here a remote robot is put into disaster response operation often controlled partly or fully by an external human commander. Usually there might be many such agents, both human and robot, internal or external. This kind of setup is typical in USAR settings (Bartlett 2015) where the robot's job is to infiltrate areas that may be otherwise harmful to humans, and report on its surroundings as and when required / instructed by the external, or required by its team. The external has a map of the environment, but this map may no longer be accurate in a disaster setting - e.g. new paths may have opened up, or older paths may no longer be available, due to rubble from collapsed structures like walls and doors. The same holds true for other team members in the loop. The robot

(internal) however, while updating its teammates, does not need to inform them of all these changes so as not to cause information overload of the commander who is usually otherwise engaged in orchestrating the entire operation, or its other teammates who are involved in completing their own tasks. This calls for an instantiation of MEGA to determine the appropriate model updates[5] to pass on to other agents in the team for a given task. A video demonstrating the scenario play out is available at https://goo.gl/BKHnSZ.

The scenario (illustrated in Figure 2), involves a robot 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. As mentioned before, the human agents' initial state (the map) may have drifted from the real map which the robot has – e.g. the agents may have confusion regarding which paths are clear and which ones are closed.

Here the external commander incorrectly believes that the path from P1 to P9 is clear and while the one from P2 to P3 is closed. The internal human agent, on the other hand, not only believes in the mistakes mentioned above but is also under the assumption that the path from P4 to P5 is untraversable. Due to these different initial states, each of these agents ends up generating a different optimal plan.

The plan expected by the external commander (marked in black in Figure 2) 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 (marked in orange). Finally, the optimal plan for the robot (marked in blue) involves the robot meeting the human at P4 on its way to P5. Through MEGA algorithm we hope to find the smallest explanation, which can explain this optimal plan to both human agents in the loop.

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, will be defined as

$\mathcal{I}^0 = \{$(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 –

$\mathcal{I}_{max} = \{$(at_P1), (at_human P10), ..., (clear_path P10 P9), (clear_path P9 P1), (clear_path P4 P5), (collapsed_path P4 P5)$\}$

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

[5]Note that, in this particular scenario, we only have differences in the initial states. To the algorithm this is identical to the general case in the model space.

For this scenario, the MEGA algorithm generates the following explanation –

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

It is interesting to note that, while the last two model changes are equally relevant for both the agents, the first change is specifically designed to help the internal human agent. The first update helps convince the human that the robot can indeed reach the goal through P4, while the next two help convince both agents as to why it is possible and why the robot should meet at P4 rather than other locations.

Discussion and Future Work

This paper presents our initial attempt at extending MRP based explanation to scenarios with incomplete human mental models or multiple explainees. We argue that in such cases, the robot should try to generate explanations that satisfy all the explainees. As pointed out in earlier sections, the algorithm introduced in this paper are quite comparable to the original model space explanation generation algorithms (Chakraborti et al. 2017b) in terms of its computational complexity. But one can easily see that the robot will need to provide a much larger explanation to satisfy the more incomplete models (either because of high uncertainty about the model or because of a larger set of explainees). One could imagine cases, where the robot might prefer to produce explanations that only work for a subset of explainees or possible models, or where a human's response to a less **robust** explanation can be quite illuminating about the human's underlying mental model.

Another exciting avenue of research is the learning of annotated models. Most of the current work on learning planning models have focused on learning complete planning model from successful plan traces (Yang, Wu, and Jiang 2005), (Cresswell, McCluskey, and West 2013). But in the case of learning mental models, such traces may be hard to come up by and even impossible. By learning annotated models, we can potentially preserve a set of possibly conflicting hypotheses and only eliminate a possible model if we can produce an observation that invalidates it. Systems that meet some of these requirements include MARSHAL (Bryce, Benton, and Boldt 2016) and CPISA (Nguyen, Sreedharan, and Kambhampati 2017). However, neither of them provide a perfect solution yet, the MARSHAL system may prove to be too intrusive (the need to observe plans, direct questions about domain model) in most HRI scenarios, while CPISA only extracts causal proofs from execution traces and does not learn an intermediate APDDL model. Ideally, we want approaches that can learn these models from a robot's plan traces labeled by humans, similar to (Zhang et al. 2017).

One of the fundamental premises of the setup discussed in the paper is that uncertainty over the human's mental model and presence of multiple humans in the loop (with known or uncertain models) is essentially equivalent in so far as the explanation generation technique is concerned. We have shown how we can address both settings with the same compilation,

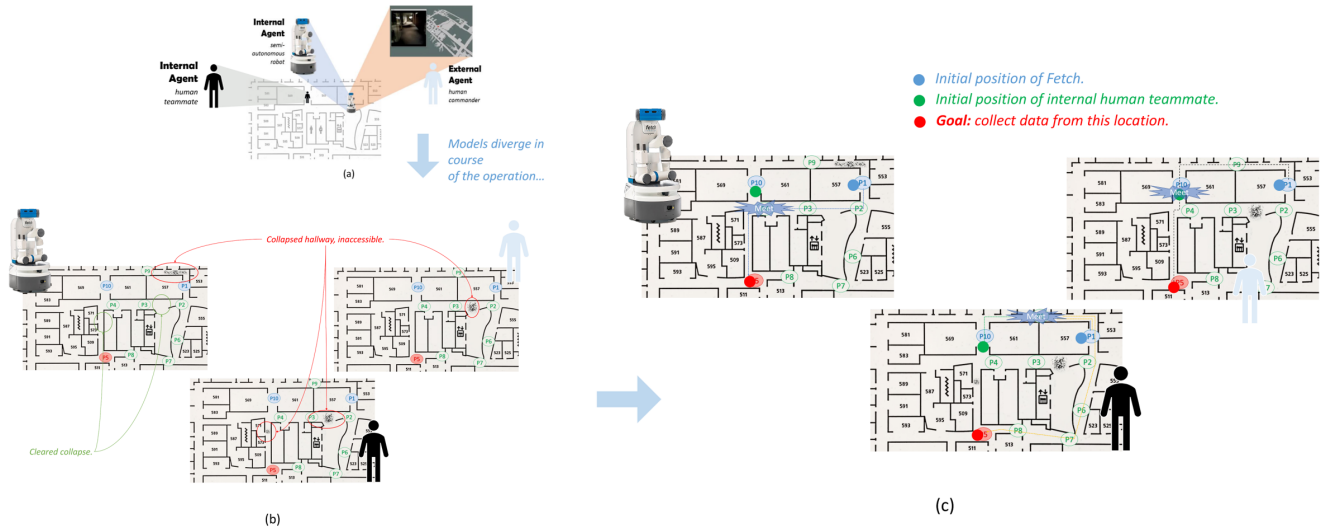

Figure 2: An USAR scenario with two human teammates and a robot. It is possible that over time, the models of the agents may diverge. In such cases, it is important that the robot can come up with explanations that satisfy all the agents involved.

and computed explanations that are valid for all possible models or all the explainees as the case may be. However, the process of explaining to the humans themselves might be different depending on the setup. For example, in the case of model uncertainty, the safest approach might be to generate explanations that work for the largest set of possible models, but 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 teaming relationship with the human.

Finally, annotated planning model is only one of the many incomplete models that have been studied in planning literature. One could choose to use an even shallower (Kambhampati 2007) planning model to reduce the model learning cost – e.g. a word vector based action affinity model (Tian, Zhuo, and Kambhampati 2016) or the CRF based plan labeling model (Zhang et al. 2017). While these models may capture human expectations and preferences about the robot plan, in terms of expressiveness of the representation they may be entirely different from human's mental model of the robot. In (Chakraborti et al. 2017a) we discuss a few such useful representations for learning such models for the purposes of task planning at various levels of granularity. If we wish to use these models, we will also need to reconsider how we can perform model reconciliation when the difference between the learned mental model and the robot model may no longer be meaningful to the human.

Conclusion

We saw how the explanation generation as model reconciliation technique can be extended to account for multiple possible models of the explainee – this is useful both in cases where the model of the explainee is uncertain as well as there are many explainees to explain to. We demonstrated such a scenario with a robot involved in a typical search and reconnaissance scenario with external supervisors whose models of the environment might have drifted in course of the operation. In (Sreedharan, Chakraborti, and Kambhampati 2017) we demonstrated how the plan explanation problem and the plan explicability problem (Zhang et al. 2017) can be treated under a single framework – we are currently developing approaches to bridge the same gap in the current context of model uncertainty / multiplicity in the context of *"model-lite"* planning (Kambhampati 2007).

Acknowledgments This research is supported in part by the ONR grants N00014161-2892, N00014-13-1-0176, N00014- 13-1-0519, N00014-15-1-2027, and the NASA grant NNX17AD06G. Chakraborti is also supported in part by the IBM Ph.D. Fellowship 2017.

References

Bartlett, C. E. 2015. Communication between Teammates in Urban Search and Rescue. *Thesis*.

Bryce, D.; Benton, J.; and Boldt, M. W. 2016. Maintaining evolving domain models. In *IJCAI*.

Chakraborti, T.; Kambhampati, S.; Scheutz, M.; and Zhang, Y. 2017a. AI Challenges in Human-Robot Cognitive Teaming. *arXiv preprint arXiv:1707.04775*.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017b. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *IJCAI*.

Cresswell, S. N.; McCluskey, T. L.; and West, M. M. 2013. Acquiring planning domain models using locm. *The Knowledge Engineering Review*.

Kambhampati, S. 2007. Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain models. In *AAAI*.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. Pddl-the planning domain definition language.

Nguyen, T. A.; Do, M.; Gerevini, A. E.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence*.

Nguyen, T.; Sreedharan, S.; and Kambhampati, S. 2017. Robust planning with incomplete domain models. *Artificial Intelligence*.

Russell, S., and Norvig, P. 2003. *Artificial intelligence: a modern approach*. Prentice Hall.

Sreedharan, S.; Chakraborti, T.; and Kambhampati, S. 2017. Balancing Explicability and Explanation in Human-Aware Planning. *ArXiv e-prints* abs/1708.00543.

Tian, X.; Zhuo, H. H.; and Kambhampati, S. 2016. Discovering underlying plans based on distributed representations of actions. In *AAMAS*.

Yang, Q.; Wu, K.; and Jiang, Y. 2005. Learning actions models from plan examples with incomplete knowledge. In *ICAPS*.

Zhang, Y.; Sreedharan, S.; Kulkarni, A.; Chakraborti, T.; Zhuo, H. H.; and Kambhampati, S. 2017. Plan Explicability and Predictability for Robot Task Planning. In *ICRA*.