Interactive Plan Explicability in Human-Robot Teaming

Mehrdad Zakershahrak, Akshay Sonawane, Ze Gong and Yu Zhang

Abstract— **Human-robot teaming is one of the most important applications of artificial intelligence in the fast-growing field of robotics. For effective teaming, a robot must not only maintain a behavioral model of its human teammates to project the team status, but also be aware of its human teammates' expectation of itself. Being aware of the human teammates' expectation leads to robot behaviors that better align with the human expectation, thus facilitating more efficient and potentially safer teams. Our work addresses the problem of human-robot interaction with the consideration of such teammate models in sequential domains by leveraging the concept of plan explicability. In plan explicability, however, the human is considered solely as an observer. In this paper, we extend plan explicability to consider interactive settings where the human and robot's behaviors can influence each other. We term this new measure Interactive Plan Explicability (IPE). We compare the joint plan generated by our approach with the consideration of this measure using the fast forward (FF) planner, with the plan generated by FF without such consideration, as well as with the plan created with human subjects interacting with a robot running an FF planner. Since the human subject is expected to adapt to the robot's behavior dynamically when it deviates from her expectation, the plan created with human subjects is expected to be more explicable than the FF plan, and comparable to the explicable plan generated by our approach. Results indicate that the explicability score of plans generated by our algorithm is indeed closer to the human interactive plan than the plan generated by FF, implying that the plans generated by our algorithms align better with the expected plans of the human during execution. This can lead to more efficient collaboration in practice.**

I. INTRODUCTION

There is an ever-growing number of robotic applications, among which many depend on the ability of the robot being an effective teammate. The team effectiveness is a compound metric that captures how team members consistently act according to the expectation of the team [1]. Effective teaming consists of (1) situation awareness in terms of recognizing the status of the team tasks and teammates' states, (2) shared mental model to predict or foresee the next action of the team under the current context, (3) direct and indirect interaction between the teammates, and (4) taking proactive actions considering other team members' subgoals to support their achievement [2], [1].

Hence, to achieve a comparable level of efficiency as in human teams, a key challenge in human-robot teaming is to ensure that the robot always assists humans in an expected and understandable fashion that is consistent with the teaming context. To do this, a robotic teammate must first be able to recognize the intent of its human teammates and then coordinate with them in a way that is expected. This argument has been more prominently made recently in human-robot teaming research [3]. It is argued that the robot must maintain mental models of its human teammates. These mental models not only include human intent and other mental states, but also their expectations of the robot. The ability to accommodate such expectations can lead to more fluent teaming, even though it often leads to suboptimal plans due to the differences between the robot's plan and that of the human expectation.

There are many reasons why the robot's plan would differ from that of the human expectation. For example, humans may misunderstand the abilities of their robotic teammates, resulting in inconsistencies between the robot's domain model and the human's interpretation of this model. Modeling the human expectation is particularly challenging since the robot often does not have direct access to it and it is difficult to be learned. To address this issue, we use the notion of plan explicability discussed in [4], where an approach was proposed to learn the model of expectation based on a labeling process. That work, however, focused on the human being an observer. In this work, we extend the notion of plan explicability to an interactive setting where the human is cooperating with the robot.

In an interactive teaming setting, the behaviors of the human and robot can influence each other. For instance, consider a scenario where a human is assigned to a first-response task with a robotic teammate after a disaster occurred. Due to the hazardous situation in the environment, the human stays at the command center and the robot enters the environment to provide medical assistance at the locations where injured people are likely to be present. The team's goal is to provide medical assistance as quickly as possible. However, due to damages incurred by the disaster, some paths may be blocked which is unknown to the human and only perceivable by the robot teammate that is working at the disaster scene. Hence, the situation may happen that the human would command the robot to visit a room and expects the robot to follow the shortest path in her view, but the robot would take a longer route due to obstacles that the human is unaware of. This robot behavior from the human's perspective is inexplicable. In an interactive setting, this may trigger the human to more closely monitor the robot's behavior and command the robot more frequently. These interactions directly influence the mental models of the human and hence can change her teaming behavior, which would in turn affect the robot, thus forming a tight interaction. In such a case, a plan is comprised of both human and robot actions, and the influence

[1] Mehrdad Zakershahrak, Akshay Sonawane, Ze Gong and Yu Zhang are with the Computer Science and Engineering Department at Arizona State University {mzakersh, apsonawa , zgong11, yzhan442}@asu.edu

of the agent's behavior on each other must be explicitly considered.

In such teaming scenarios, the ability of the robot to predict the joint team behavior hence can increase the team effectiveness since the robot can now anticipate the human's response and how it should react accordingly. This in turn allows the robot to choose plans that are the least interruptive to the human thus improving teaming fluency. To achieve this, similar to plan explicability [4], we assume that humans interpret robot plans by attaching abstract task labels to robot actions as a labeling process. The difference here is that the plan contains not only robot actions but also human actions. The human actions provide the teaming context for the labeling process, which is modeled using Conditional Random Fields (CRFs). The learned model can be used to label a new team plan to compute its interactive plan explicability score, similar to the explicability score in [4]. Having these measures allows the robot to synthesize plans that are more explicable to the human. Our contribution in this work includes extending plan explicability to interactive teaming scenarios, implementing a plan monitoring and replanning process during actual human-robot interaction, as well as evaluating this approach using a synthetic first response domain.

II. Related Work

The notion of robotic teammate, or that using robots to complement humans in various tasks, has attracted lots of research interest. At the same time, however, the realization of this notion is challenging due to the human-aware aspect [3], or that the robot must consider the human in the loop, both in terms of physical and mental models while planning to achieve the team goal. In such cases, it is no longer sufficient to model humans as parts of the environment [5]. Instead, human-robot teaming applications require the robot to be proactive in assisting humans [6].

There are different aspects to be considered for human-robot teaming. First, the robot must take the human's intent into account. Various plan recognition algorithms [7], [8] can be applied to perform plan recognition based on a given set of observations. The challenge is how the robot can utilize this information to synthesize a plan while avoiding conflicts or providing proactive assistance [9], [10]. There are different approaches to planning with such consideration [5], [11].

A more challenging aspect, for the robot to be considered as a teammate, is to be socially acceptable, where the robot must be aware of the expectation of the human teammates and act accordingly. The challenge is to model the human's expectation of the robot and align the robot's behavior with this expectation. In [12], the approach is to generate "legible" motions that show the robot's intent implicitly [13]. Another approach is to train the team sufficiently so that each team member would maintain a good prediction model of each other's behavior [14]. These approaches, however, work only in relatively simple and repetitive domains. For more complex domains, the robot is required to learn and model the human expectation from interactions [3], [15]. Using these models, the robot will be able to anticipate human expectations in order to remain comprehensible to the human, or to choose a behavior that is the least interruptive when it does not match perfectly with the expectation. This ability is well known to promote sustainability of teaming situation awareness [2] in human-human teams. While this work is inspired by [4], we significantly extend the framework to consider interactive human-robot teaming instead of having the human being merely an observer.

III. Background

A. Planning

A planning problem can be formulated as a tuple P = $\langle F, M, I, G\rangle$, where F is a set of fluents, M is the domain model which consists of a set of actions A and a cost function C. $I \subseteq F$ is the initial state and $G \subseteq F$ is the goal state. Each action in A is a tuple consists of preconditions and effects. C assigns a non-negative cost to each action. Given a planning problem with I and G, the objective is to synthesize a plan $\pi = \langle a_1, a_2, ..., a_n\rangle$ which consists of a sequence of actions that lead to the goal state from the initial state. The cost $c(\pi)$ is the sum of the costs of all the actions in the plan π.

B. Plan Explicability

The explicability of a plan [15] is correlated with a mapping of high-level tasks (as interpreted by humans) to the actions performed by the robotic agent. The demand for generating explicable plans is due to the inconsistencies between the robot's model and the human's interpretation of the robot model (which captures the human's expectation of the robot). To formalize the explicable planning problem, consider the setting with two models where M_R is the robot model and $\widetilde{M_R}$ is the human's interpretation of M_R. For a given initial and goal state pair, $\langle I, G\rangle$, let π_{M_R} be a plan generated by the robot using M_R, and $\pi_{\widetilde{M_R}}$ be the plan of the human's expectation using $\widetilde{M_R}$. An explicable plan in M_R is a plan π_{M_R} that minimizes the weighted sum of plan cost of π_{M_R} and the plan distance between π_{M_R} and $\pi_{\widetilde{M_R}}$. It can be written as:

$$argmin_{\pi_{M_R}} cost(\pi_{M_R}) + \alpha \cdot dist(\pi_{M_R}, \pi_{\widetilde{M_R}}) \qquad (1)$$

where *cost* returns the cost of a plan, *dist* computes the distance between two plans, and α denotes the relative weight. In Eq. (1), $\widetilde{M_R}$ is often unknown and *dist* needs to be specified. To deal with this, in [15], the distance between the two plans is approximated using a CRF model, where a labeling scheme is used to map the human interpretations of the robot's actions as task labels to the robot actions in π_{M_R}. Then the *dist* function is defined as a composition of two functions as shown in Eq. (2), where F is a domain independent function that takes plan labels as its input and $L^\star(\pi_{M_R})$ is a labeling scheme that maps task labels to the actions in M_R.

$$dist(\pi_{M_R}, \pi_{\widetilde{M_R}}) = F \circ L^\star(\pi_{M_R}) \qquad (2)$$

Using a CRF model to learn the labeling scheme $L^\star$, Eq. (2) becomes:

$$argmin_{\pi_{M_R}} cost(\pi_{M_R}) + \alpha \cdot F \circ L_{CRF}(\pi_{M_R} \mid \{S_i \mid S_i = L^\star(\pi^i_{M_R})\}) \quad (3)$$

where $L_{CRF}(\pi_{M_R})$ is the learned CRF model of $L^\star$ and $\{S_i\}$ is the training data.

Plan Explicability: Given a robot plan π in M_R

$$\pi = \langle a_0, a_1, a_2, ..., a_N \rangle \quad (4)$$

where a_0 is the starting action and there are N actions in π, and a set of action labels T given by

$$T = \{T_1, T_2, ..., T_M\} \quad (5)$$

where M is the number of labels, we can first apply $L^\star_{CRF}$ to obtain the label sequence, L_π. The explicability score of π is computed based on L_π. The explicability measure as in [4] is defined as follows:

$$F_\theta(L_\pi) = \frac{\sum_{i \in [1,N]} 1_{L(a_i) \neq \emptyset}}{N} \quad (6)$$

where $F_\theta(L_\pi) : L_\pi \rightarrow [0,1]$ (with 1 being the most explicable), 1 is an indicator function, and F_θ is the domain independent function that converts plan labels to the final score. When the labeling process can't assign a label to an action a_i, its label $L(a_i)$ will be the empty set (implemented as a special label).

IV. INTERACTIVE PLAN EXPLICABILITY

In our work, the robot creates composite plans for both the human and robot using an estimated human model and the robots model, which can be considered as its prediction of the joint plan that the team is going to perform. At the same time, however, the human would also anticipate such a plan to achieve the same task, except with an estimated robot model and the humans own model.

Each problem in this domain can be expressed as a tuple $P_T = \langle I, M_R, \widetilde{M_H}, \Pi_C, G \rangle$. In this tuple, I denotes the initial state of the planning problem, while G represents the shared goal of the team. M_R represents the actual robot model and $\widetilde{M_H}$ denotes the approximate human model provided to the robot, which may also be learned [16]. The actual human model M_H could be quite different from $\widetilde{M_H}$ provided to the robot. Similarly, the approximate robot model from the human $\widetilde{M_R}$ may be different from the actual robot model M_R. See an illustration of the problem setting in Fig. 1. Finally Π_C represents a set of annotated plans that are provided as the training set for the CRF model.

A. Problem Formulation

In this work, the plan for the team will be represented by a composite plan, which is defined as follows:

Definition 4.1 (Composite Plan): A composite plan π_c captures the actions performed by both the human and robot to achieve the goal and is represented as $\pi_c = \{a_1^{\phi_1}, a_2^{\phi_2}, ..., a_i^{\phi_i}, ..., a_n^{\phi_n}\}$. Here $a_i^{\phi_i}$ represents the i^{th} action in the plan performed by the agent ϕ_i (where ϕ_i can be either H or R).

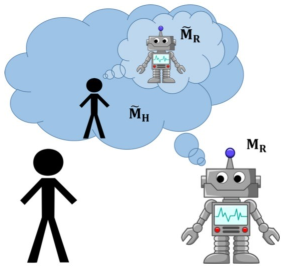

Fig. 1. The robot's planning process is informed by an approximate human model and the robot's own model, while the human's planning process is informed by an approximate robot model and the human's own model.

In our current setting, we assume that only one agent is executing its action at any given time (please see discussion section to see how we plan to relax this assumption). To generate an explicable plan, the robot needs to synthesize a composite plan that is as close as possible to the plan that the human expects. This is an especially daunting challenge, given that we have multiple points of uncertainty (e.g., $\widetilde{M_H}$ and $\widetilde{M_R}$). Nevertheless, a similar method to [4] can be utilized here by updating Eq. (1) as follows:

$$argmin_{\pi_C^{M_R,\widetilde{M_H}}} cost(\pi_C^{M_R,\widetilde{M_H}}) + \alpha \cdot dist(\pi_C^{M_R,\widetilde{M_H}}, \pi_C^{\widetilde{M_R},M_H}) \quad (7)$$

where $\pi_C^{M_R,\widetilde{M_H}}$ is the composite plan created by the robot using M_R and $\widetilde{M_H}$, while $\pi_C^{\widetilde{M_R},M_H}$ is the composite plan that assumed to be expected by the human. Similar to our prior work, we assume that the distance function $dist(\pi_C^{M_R,\widetilde{M_H}}, \pi_C^{\widetilde{M_R},M_H})$ can be calculated as a function of action labels for $\pi_C^{M_R,\widetilde{M_H}}$:

$$argmin_{\pi_C^{M_R,\widetilde{M_H}}} cost(\pi_C^{M_R,\widetilde{M_H}}) + \alpha \cdot F \circ L_{CRF}(\pi_C^{M_R,\widetilde{M_H}} \mid \{S_i \mid S_i = L^\star(\pi_C^i)\}) \quad (8)$$

Similarly, the labeling process for each action is modeled by a CRF L_{CRF} trained on a set of labeled team execution traces ($\{\pi_C^i\}$). For planning, we can easily adopt any state space planner that uses forward search, while ensuring that the heuristic itself takes into account the explicability score. To search for an explicable plan, we use a heuristic search method as shown in Algorithm 2; the heuristic is $f = g + h$, where g is the cost of the plan prefix and h is calculated as follows:

$$h = (1.0 - F_\theta(L(s.path\#rp))) * |s.path\#rp| * |rp| + |rp| \quad (9)$$

where s above is the current state, $\#$ means concatenation and $rp = relaxedPlan(s, Goal)$. The planner algorithm is provided in Algorithm 1 and the algorithm to calculate the f value is given in Algorithm 2.

B. Monitoring & Replanning for Interactive Teaming

In an interactive setting, given that the robot does not have access to the complete and accurate human model nor the human's expectation of its own model, the robot will rely on replanning when the human deviates from its plan. This is discussed in more detail next. The main components of our monitoring & replanning system for training the CRF model are as follows:

- Controller: The service controlling robot actions and the planner used by the robot to achieve the goal is presented in Algorithm 3, it starts with an initial plan and performs replanning whenever the actual human action does not align with the explicable plan.
- Planner: This module is responsible for generating the composite plans. It takes the current state, combined robot and human planning model (where the human model is an approximation of the exact model), the trained CRF model, and any plan prefix. Robot calls the planner before starting any execution. If the executed plan deviates, the controller calls the planner again with updated current state and a plan prefix consisting of all actions that have been executed up to that point. The details about the planner are covered in Algorithm 1.

```
Input: StartState, CombinedModel, Goal, PlanPrefix
CurrentState:= StartState;
CurrentPlan := Planner(CurrentState, CombinedModel,
Goal, PlanPrefix);
statePriorityQueue.add(allNeighboursWithFValue
(currentState, Goal, PlanPrefix));
while statePriorityQueue is not empty do
    currentState := statePriorityQueue.getBestState();
    if currentState satisfies Goal then
        return currentState.path;
    else
        statePriorityQueue.add(allNeighboursWithFValue
        (currentState, Goal, PlanPrefix));
    end
end
```

Algorithm 1: Algorithm for a planner to generate explicable plans

```
Input: CurrentState, Goal, PlanPrefix
neighborList := [];
for state in CurrentState.neighbors do
    rp := state.relaxedPlan(Goal);
    h := findExplicabilityScore(PlanPrefix +
    CurrentState.path + rp);
    g := CurrentState.path.cost;
    f := g + h;
    neighborList.add(tuple(state, f));
end
return neighborList;
```

Algorithm 2: Algorithm for allNeighboursWithFValue to calculate the f value for each of the neighboring states.

The controller service runs a monitoring component, which ensures that the human performs the expected action. If the state changes do not correspond to the expected action, the monitor calls the planner again to produce a new plan (replanning process). The controller feeds the planner the latest state along with the list of actions that have been executed till that point (referred to as current plan prefix). To allow the CRF model to incorporate plan context from previously executed actions, it needs to consider current plan prefix as part of a larger plan containing the previously executed actions, rather than a new planning problem.

```
Input: CombinedModel, Goal
CurrentPlan := Planner(CurrentState, CombinedModel,
Goal);
CurrentPlanPrefix := [];
for action in CurrentPlan do
    if action is robot action then
        execute action;
        add action to CurrentPlanPrefix;
    else
        executedAction := waitForHumanAction();
        add executedAction to CurrentPlanPrefix;
        if executedAction = action then
            Continue;
        else
            CurrentState := Monitor();
            CurrentPlan := Planner(CurrentState,
            CombinedModel, Goal, CurrentPlanPrefix);
        end
    end
end
```

Algorithm 3: Algorithm for the controller service.

V. Evaluation

To evaluate our system, we tested it on a simulated first response domain, where a human-robot team is assigned to a first-response task after a disaster occurred. In this scenario, the human's task is to team up with a remote robot that is working on the disaster scene. The team goal is to search all the marked locations as fast as possible and the human's role is to help the robot by providing high-level guidance as to which marked location to visit next. The human peer has access to the floor plan of the scene before the disaster. However, some paths may be blocked due to the disaster that the human may not know about; the robot, however, can use its sensors to detect these changes. Due to these changes in the environment, the robot might not take the expected paths of the human.

For data collection, we implemented the discussed scenario by developing an interactive web application using MEAN (Mongo-Express-Angular-Node) stack.

In our setting, the robot would always follow the human's command (i.e., which room to visit next). The human can, of course, change the next room to be visited by the robot anytime during the task if necessary, simply by clicking on

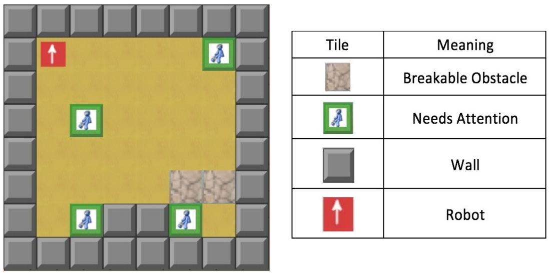

Fig. 2. A sample map that the human subjects see with a description of the object types.

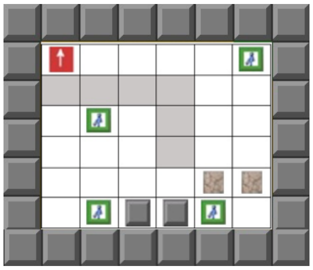

Fig. 3. A sample map corresponding to the map in Figure 2 that the robot sees; the gray cells are hidden obstacles.

any of the marked locations. The robot uses BFS search to plan to visit the next room. After a room is visited, the human cannot click on the room anymore. The robot always waits 1 second before performing the next action. For simplicity, the costs of all human and robot actions are the same.

A. *Experimental Setup*

For training, after each robot action, the system asks the human whether the robot's action makes sense or not. If the human answers positively, that action is considered to be explicable. Otherwise, the action is considered to be inexplicable. This is used later as the labels for learning the model of interactive plan explicability. All scenarios were limited to four marked locations to be visited, with a random number $(2-5)$ of visible obstacles and manually inserted hidden obstacles (invisible to the human) in the map. We have generated a set of 16 problems for training and 4 problems for testing.

We collected in total 34 plan traces for training, which were used to train our CRF model. All training data was collected with human trials, with random initial robot initial and goal locations. To remove the influence of symbol permutation, we performed the following processing on the training set: For each problem, we created an additional 1000 traces that are the same problem only with different permutations of symbols.

A sample map of the actual environment is shown in Figure 2. Figure 3 shows the same map that the robot sees with hidden obstacles drawn on the map.

B. *RESULTS*

Table I shows the ratios (refer to as the *explicability ratio*) between the number of explicable actions and the number of actions over all plans, created for the testing problems using our approach, FF planner, and human plan, respectively. The interactive explicable plan (our approach) is created using the heuristic search method mentioned in Equation (9). Note that all the human actions will be considered explicable in our plans (although one can argue that is not the case).

As we can see in Figure 4, the explicability ratio for our approach is similar (0.1% difference) to the human plan while being quite different from the FF plan (13.9% difference). This is also intuitively explained in Fig. 4, where We can clearly see that the explicable plan is similar to the human plan, in the sense the human tends to change commands in this task domain due to unknown situation.

The above results show that the plans created by our algorithm are closer to what the human expects, and thus enabling the robot to better predict the team behavior and potentially lead to more efficient collaboration in practice. The explicability scores for the four testing problems are shown in Table II. The reason for the low explicability score of FF plan is that FF tends to create plans that are less costly while ignoring the fact that the human and robot may view the environment and each other differently, and thus less costly plans in one view are also more likely to be misaligned with less costly plans in the other. Note, however, that whether the explicable plan would lead to better teaming performance (e.g., less replanning efforts for the robot and less cognitive load for the human) requires further investigation and evaluation with actual human subjects. This will be explored in future work.

TABLE I
COMPARISON OF EXPLICABILITY RATIO FOR TESTING SCENARIOS

Plan Type	Interactive Explicability Score
Interactive Explicable Plan	0.820
FF Planner	0.672
Human Plan	0.811

TABLE II
ELABORATED EXPLICABILITY SCORE FOR TEST SCENARIOS

Scenario #	FF Plan	Interactive Explicable Plan
1	1.0	1.0
2	0.56	0.714
3	0.629	0.757
4	0.8	0.8

VI. CONCLUSIONS AND FUTURE WORK

We created a general way of generating explicable plans for human-robot teams, where the human is an active player. This differs from prior work in the sense that we do not assume that the human and robot have the same knowledge about the environment and each other; or in other words, there exists information asymmetry, which is often true in realistic task domains. To generate an explicable plan for a human-robot team, we need not only consider the plan

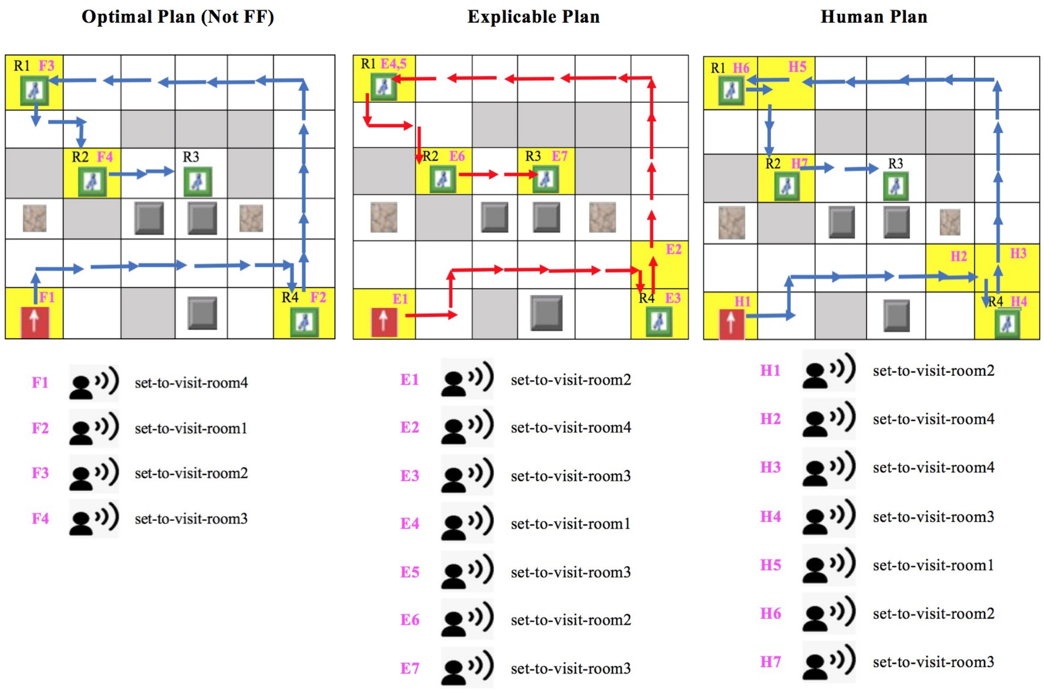

Fig. 4. Comparison of plans for a specific problem. (Left) The optimal plan; (Middle) The explicable Plan; (Right) The human plan. The initial location of the robot is indicated with a white arrow inside a red box. Yellow cells refers to where the human commands are received.

cost, but also the preconceptions that the human may have about the robot. Although we have mainly focused on two member teams, we believe that these ideas can be easily extended to larger team sizes with a few changes to the current formulation. It should also be straightforward to extend the current formulation to support simultaneous action executions by considering joint actions at any time step. Another way we may be able to achieve this would be by using temporal planners [18] instead of relying on sequential ones. Also, the current system assumes the provision of an approximate human planning model and relies on replanning to correct its plans whenever the human deviates from the predicted explicable plan. We could possibly explore the idea of incorporating models like capability model [16] to learn such human models.

References

[1] N. R. Council, *Enhancing the Effectiveness of Team Science*, N. J. Cooke and M. L. Hilton, Eds.

[2] N. J. Cooke, "Team cognition as interaction," *Current directions in psychological science*, vol. 24, no. 6, pp. 415–419, 2015.

[3] T. Chakraborti, S. Kambhampati, M. Scheutz, and Y. Zhang, "Ai challenges in human-robot cognitive teaming," *arXiv preprint arXiv:1707.04775*, 2017.

[4] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, and S. Kambhampati, "Plan explicability and predictability for robot task planning," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1313–1320.

[5] T. Chakraborti, G. Briggs, K. Talamadupula, Y. Zhang, M. Scheutz, D. Smith, and S. Kambhampati, "Planning for serendipity," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 5300–5306.

[6] A. Fern, S. Natarajan, K. Judah, and P. Tadepalli, "A decision-theoretic model of assistance." in *IJCAI*, 2007, pp. 1879–1884.

[7] H. A. Kautz and J. F. Allen, "Generalized plan recognition." in *AAAI*, vol. 86, no. 3237, 1986, p. 5.

[8] M. Ramırez and H. Geffner, "Probabilistic plan recognition using off-the-shelf classical planners," in *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI 2010)*, 2010, pp. 1121–1126.

[9] T. Chakraborti, Y. Zhang, D. E. Smith, and S. Kambhampati, "Planning with resource conflicts in human-robot cohabitation," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 1069–1077.

[10] M. Cirillo, L. Karlsson, and A. Saffiotti, "Human-aware task planning for mobile robots," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–7.

[11] T. Chakraborti, S. Sreedharan, Y. Zhang, and S. Kambhampati, "Plan explanations as model reconciliation: Moving beyond explanation as soliloquy," in *Proceedings of IJCAI*, 2017.

[12] A. Dragan and S. Srinivasa, "Generating legible motion," in *Robotics: Science and Systems*, June 2013.

[13] R. A. Knepper, C. I. Mavrogiannis, J. Proft, and C. Liang, "Implicit communication in a joint action," in *Proceedings of the 2017 acm/ieee international conference on human-robot interaction*. ACM, 2017, pp. 283–292.

[14] S. Nikolaidis, K. Gu, R. Ramakrishnan, and J. Shah, "Efficient model learning for human-robot collaborative tasks," *arXiv preprint arXiv:1405.6341*, 2014.

[15] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, and S. 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.

[16] Y. Zhang, S. Sreedharan, and S. Kambhampati, "Capability models and their applications in planning," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1151–1159.

[17] J. Hoffmann and B. Nebel, "The FF planning system: Fast plan generation through heuristic search."

[18] M. B. Do and S. Kambhampati, "Sapa: A multi-objective metric temporal planner," *J. Artif. Intell. Res.(JAIR)*, vol. 20, pp. 155–194, 2003.

[19] Y. Zhang, H. H. Zhuo, and S. Kambhampati, "Plan explainability and predictability for cobots," *CoRR abs/1511.08158*, 2015.