Robust Planning for Human-Robot Joint Tasks with Explicit Reasoning on Human Mental State

Anthony Favier, Shashank Shekhar, Rachid Alami

Robust Planning for Human-Robot Joint Tasks with Explicit Reasoning on Human Mental State

Anthony Favier[1,2], **Shashank Shekhar**[1], **Rachid Alami**[1,2]

[1]LAAS-CNRS, Universite de Toulouse, Toulouse, France
[2]Artificial and Natural Intelligence Toulouse Institute (ANITI) {anthony.favier, sshekhar, rachid.alami}@laas.fr

Abstract

We consider the human-aware task planning problem where a human-robot team is given a shared task with a known objective to achieve. Recent approaches tackle it by modeling it as a team of independent, rational agents, where the robot plans for both agents' (shared) tasks. However, the robot knows that humans cannot be administered like artificial agents, so it emulates and predicts the human's decisions, actions, and reactions. Based on earlier approaches, we describe a novel approach to solve such problems, which models and uses *execution-time observability conventions*. Abstractly, this modeling is based on situation assessment, which helps our approach capture the evolution of individual agents' beliefs and anticipate belief divergences that arise in practice. It decides *if* and *when* belief alignment is needed and achieves it with communication. These changes improve the solver's performance: (a) communication is effectively used, and (b) robust for more realistic and challenging problems.

Introduction

Consider a scenario where you (the *human* agent) want to prepare, together with a robot, some pasta in your kitchen. This joint task comprises several *components* and *activities*, e.g., pasta is kept either in the kitchen or the living room, covering a pot, turning a furnace on, the salt container, adding salt to the pasta, etc. Sometimes, some components and activities might be accessible only to you (the robot) from your (its) current position, while some are accessible to both. Like this subclass of joint task, its multiple instantiations exist across many real-world domains, e.g., homes, offices, hospitals, etc.

Naturally, while achieving such tasks, you would want a pleasant and flexible collaboration. This involves behaviors like not being bothered too much by the robot or being able to delegate tasks to it, even if it delays the overall process. Here, we consider the human agent cooperative and rational but cannot be administered like a robot (a *controllable* agent). So, if the human agent has several ways to achieve a goal or accomplish a task, the robot cannot instruct them explicitly which one to choose. However, it can still act to influence their choices, thus eliciting future actions.

Presented at the AI-HRI Symposium at AAAI Fall Symposium Series (FSS) 2022

The above example outlines an important point about human-robot collaboration. The robot should not only plan for agents' joint actions but also predict and emulate human decisions, actions, and reactions, to achieve a joint goal seamlessly. It must put itself in the shoes of the human in order to better predict their decisions based on an accurate estimation of their beliefs; this would open the possibility for the robot to "create" circumstances that promote actions to be performed by humans and relevant to achieving the joint task or to prevent them from making errors based on false, outdated or inaccurate beliefs.

Human-aware planning has been a topic of interest, and a framework proposed and upgraded over the years is called HATP (Human-Aware Task Planner) (Alami et al. 2006; Alili, Alami, and Montreuil 2009; Lallement, De Silva, and Alami 2014; Lallement, de Silva, and Alami 2018). Precisely, the HATP framework extends the existing Hierarchical Task Network (HTN) framework (Ghallab, Nau, and Traverso 2004), extending its representation and semantics to make it more suitable for robotics and, more specifically, human-robot collaboration. The agents become the first class entities while "social rules" are added to classify planned behaviors for the robot as (un)acceptable.

A more elaborate planning architecture, inspired by HATP although extends HTNs differently, has been recently proposed (Buisan and Alami 2021; Buisan et al. 2022). It introduced the *first* planning framework of this type, discussing the abilities and framework's broad spectrum. It is called HATP/EHDA, which extends HATP by **e**mulating **h**uman **d**ecisions and **a**ctions. The new planning scheme raises several interesting, non-trivial questions to be investigated, and their principled will improve the proposed HATP/EHDA framework toward task-level autonomy for human-robot interaction or human-robot collaboration.

HATP/EHDA comprises a *dual*-HTN structure that can be seen as a joint hierarchical task specification model. We assume such specifications are given by a domain modeler. The joint model describes agents' capabilities, initial beliefs, shared tasks, world dynamics, understanding of common ground, etc. Moreover, the modeler implicitly captures hypothetical variables to represent the human mood, intentions, etc., that are non-trivial to manage explicitly but "affect" the human behavior.

HATP allows only one search thread for a shared plan, re-

sulting in two coordinated plan streams for the robot and human team. However, HATP/EHDA proposed a two-threaded search process, generating a policy tree comprising both agents' actions. The human agent is *uncontrollable*, so it is not trivial to determine their exact "next" feasible action for the overall joint task. It captures the impact of their mental state, mood, intention, etc. Also, it is important to note that both the robot's and human's HTNs are within the robot. At each stage, it must plan their joint actions following the specifications while predicting, managing, and emulating human's decisions, goals, beliefs, etc.

However, the current implementation of the HATP/EHDA framework makes some simplistic assumptions. E.g., an action performed by an agent would also impact the beliefs of other agents, i.e., considering that other agents are always aware of the action execution and share the same perspective as the actor. In reality, a robot's action can influence human's beliefs differently under different scenarios. For example, adding salt to the pasta or switching on the furnace (in the kitchen) would impact the human's belief state differently depending on whether the human is also in the kitchen or not at the time of action execution.

We introduce a novel paradigm for implementing *suitable conventions* understood during plan execution in this context. This paradigm helps upgrade the HATP/EHDA's planning system. A high-level idea to formalize it is inspired by existing situation assessment reasoners which perform spatial reasoning and perspective taking (Flavell 1992; Trafton et al. 2005; Johnson and Demiris 2005; Sisbot, Ros, and Alami 2011; Warnier et al. 2012; Lemaignan et al. 2017). The upgraded planning system handles divergences in human-robot individual beliefs (computed using our proposed model) and plans with explicitly modeled communication actions such that it tackles belief divergences via communication, if and when needed.

Effective use of a communication modality is essential to achieve the motivation behind their development, say, a seamless collaboration. However, in reality, communication incurs some costs. Moreover, in human-robot interaction (HRI), communicating too much or too little can disturb the overall task achievement process. Hence, deciding "how" to communicate is crucial, but it is equally crucial to decide, "if", "when", and "what" to communicate. Assume that a *speech* modality is available while the *decisional* aspect of communication is handled by the new solver with an enhanced reasoning system on the human mental state.

Related Work

Several attempts to model human activities are made such that these models are used to design a system, which integrates into another paradigm dealing with human tasks to improve the latter's performance.

A common way to represent human activity and interaction with a computer at an abstraction level is by using *task models*. The hierarchical structure of activities was first exploited by Annett and Duncan (1967), and they said a task could be seen at several abstraction levels until some criterion is satisfied. Each can, thus, refine into more concrete subtasks detailing the procedure followed by the human to achieve the higher-level task. Task modeling evolved to introduce system interactions, produced and needed information, potential errors, and a variety of operators specifying how tasks interact with each other during their execution. Task models' usage is common in user-centered and user interface designing processes. Most advanced notations include CONCURTASKTREES (Paternò 2004) and HAMSTERS (Martinie et al. 2019). Such models are used for designing or evaluating interactive systems, provide a task understanding, and inspire this work since they share a high-level architecture with HTNs.

Previous approaches are based on decomposing tasks hierarchically to human-aware task planning and assume a fully controllable and cooperative human agent willing to participate in achieving a joint-goal (Alami et al. 2006; Alili et al. 2009; Lallement, De Silva, and Alami 2014; De Silva, Lallement, and Alami 2015; Lallement, de Silva, and Alami 2018). Moreover, the agents are assumed to have established a shared goal before planning. Later, the generated plan gets shared with the human before the execution (Milliez et al. 2016). We note that HATP does not represent humans as *regular* agents with separate decision processes, which may lead to diverging plans without robot communication.

Other approaches are distantly related to what we do, consider an external human model (e.g., Agent Markov Models (AMMs) (Unhelkar, Li, and Shah 2020a, 2019)), which predicts human activities, and hence the robot plans accordingly (Hoffman and Breazeal 2007; Unhelkar, Li, and Shah 2020a, 2019). Such systems can determine actions that influence human future actions. While some systems interact with humans even when they are non-collaborating (Buckingham, Chita-Tegmark, and Scheutz 2020).

In human psychology, especially in human-human interaction or collaboration, spatial reasoning and perspective-taking are crucial (Flavell 1992; Tversky, Lee, and Mainwaring 1999). Through them, a person can mimic the mind of another person to understand their point of view. Indeed, Theory of Mind (ToM) refers to the ability to ascribe distinct mental states to other people and update them by reasoning about their perceptions and goals (Premack and Woodruff 1978; Baron-Cohen, Leslie, and Frith 1985).

Ideas from psychology got employed in many human-robot contexts. In robotics, ToM often focuses on perspective taking and belief management such that a robot reasons out what humans can perceive (Berlin et al. 2006; Milliez et al. 2014) and builds representations of the environment from their perspective, which sometimes helps solve ambiguous tasks, predict human behavior, etc. Johnson and Demiris (2005) propose a method based on visual perspective taking for a robot recognizing an action executed by another robot. While Milliez et al. (2014) propose visual and spatial perspective taking to find out *the referent* indicated by the human. In (Sisbot, Ros, and Alami 2011), based on spatial reasoning and perspective taking, a reasoner generates online (symbolic) relations between agents and objects *co-present* in the environment. Such relations are stored in databases, allowing access to the complete framework, and used for planning, acting, or both.

The framework proposed in (Devin and Alami 2016) al-

lows the robot to estimate the mental state of the human, which contains not only the human's belief but their actions, goals, and plans. It supports the robot's capabilities to do spatial reasoning w.r.t. the human and track their activities. In particular, it shows how to manage the execution of shared plans in a collaborative object manipulation context and how a robot can adapt to human decisions and actions while communicating when needed.

This work is motivated by the main ideas of ToM (Devin and Alami 2016) and situation assessment based on spatial reasoning and observability while abstractly employing them during planning.

Communication is a key to successful human-robot collaboration, used to align an agent's belief, clarify its decision or action, fix errors, etc. (Tellex et al. 2014; Sebastiani et al. 2017). We already discussed the ToM-enabled framework by Devin and Alami (2016). The framework handles execution time subtleties such that it decides at the execution time if communication is needed and then the content that should be transmitted. They achieve it by monitoring the divergence between the robot knowledge and the estimated human knowledge. If a belief divergence is detected that can endanger the plan, then verbal communication takes place. Recent work deals with an explicit usage of communication actions in planning (Buisan, Sarthou, and Alami 2020; Nikolaidis et al. 2018; Roncone, Mangin, and Scassellati 2017; Sanelli et al. 2017; Unhelkar, Li, and Shah 2020b). E.g., in (Roncone, Mangin, and Scassellati 2017; Unhelkar, Li, and Shah 2020b), the authors represent and plan with explicit communication actions, considering them as regular POMDP actions, such that execution policies generated contain communication actions.

However, this work estimates the evolution of the agents' beliefs and decides "if" and "when" *belief alignment* is necessary for planning. And, it is achieved via explicit communication actions, answering "what" to communicate. Our solver does not use these actions (for the deliberation process) like (non-) primitive tasks but to minimally align an agent's belief with the ground reality so that the belief divergence does not impact the overall task achievement. At each stage, if needed, a *sequence* of communication actions is *computed* by a modified Breadth-First Search planning subroutine and *appended* to the sender's plan. We provide clarification on this when we formalize our contributions.

Relevant Background

Hierarchical Task Networks

Generalized basic terminologies and definitions related to HTNs are presented, e.g., task network, problem, and solution definitions (Ghallab, Nau, and Traverso 2004).

A *task network* is a 2-tuple $w = (U, C)$, where $u \in U$ is a task node, while t_u is the task associated to this node. C is the set of constraints - including orderings between task nodes, binding constraints, etc. If $\forall u \in U$, t_u is a primitive task, the task network is primitive; otherwise, non-primitive.

Definition 1. ***(HTN Planning Problem.)*** *The HTN planning problem is a 3-tuple $\mathcal{P} = (s_0, w_0, D)$ where s_0 is the initial belief state (the ground truth), w_0 is the initial task network, and D is the HTN planning domain which consists of a set of tasks and methods.*

A *domain* is a 2-tuple $D = (O, M)$ where O is the set of operators and M is the set of methods. An operator $o \in O$ is a primitive task described as $o = (name(o), pre(o), \textit{eff}(o))$, i.e., its *name*, *precondition*, and *effect*, respectively.

A *task* consists of a task symbol and a list of parameters. It is called a *primitive* task if its task symbol is an operator name and its parameters match, otherwise, it is a *non-primitive* task. A primitive task is assumed to be directly *executable*, while to execute a non-primitive task, we need to decompose it into sub-tasks using *methods*.

A method ($m \in M$) is a 4-tuple $m = (name(m), task(m), subtasks(m), constr(m))$ which are method name, a non-primitive task, a set of sub-tasks, and a set of constraints, respectively. m is relevant for a task t, if for a parameter substitution σ, $\sigma(t) = task(m)$. There could be different methods relevant for a single task t, decomposing it differently, depending on the context. $(subtasks(m), constr(m))$ is a task network in m.

Suppose m is an instantiated method, and $task(m) = t_u$ then m decomposes u into $subtasks(m')$, producing a new task network, $\delta(w, u, m) = ((U - \{u\}) \cup subtasks(m'), C' \cup constr(m'))$. Here, every *precedence*, *before*, *after*, and *between* constraints between tasks are carefully maintained after each decomposition.

Definition 2. ***(Solution Plan.)*** *A sequence of primitive actions $\pi = (o_1, o_2, o_3..., o_k)$ is a solution plan for the HTN planning problem $\mathcal{P} = (s_0, w_0, D)$ iff there exists a primitive decomposition w_p (of the initial task network w_0), and π is an instance of it.*

The HATP/EHDA Framework

The framework comprises a dual-HTN based joint-task specification model. For ease of exposition, consider a team of a single human and a single robot. Two categories: the robot model, HTN_r, represents the task specifications for the controllable agent, while HTN_h, represents the model for the human, who is an uncontrollable one but the planner has its model representation and relevant interpretations to predict and emulate their decisions, actions, and reactions.

We restrict the framework's description to what is relevant but briefly discuss its ability to capture a broad class of scenarios. It supports the robot to act in the presence of a human agent even when they do not share a task to achieve in the beginning. Or, it can also support the robot planning for both agents by asking the human to help it occasionally, can manage the creation of shared tasks, handle human reactions modeled explicitly via triggers, etc. More details on triggers in (Ingrand et al. 1996; Alami et al. 1998).

A brief working description of the HATP/EHDA framework is as follows. The structure manipulated by it is "agent" (Buisan 2021; Buisan et al. 2022). Two such structures are used, one for the *human* and one for the *robot*, each includes a model of the corresponding agent. Each model (structure) has its own belief state, action model, task network, plan, and triggers. We are interested in solving the HATP/EHDA problem, $\mathcal{P}_{hr}$, in which agents start with a

joint task, i.e., a shared initial task network, w_0, to decompose. The existing planner uses agents' action models and beliefs to decompose the given task network into its legal primitive components. Decomposition updates the current network by inserting new (non-) primitive tasks, additional constraints, etc., such that the single-agent process is generalized for the two-agent scenario. While doing so, it also updates the belief state of each agent and models their reaction by executing the triggers.

A simplifying assumption is that agents *act* alternatively, while only one is *allowed* to act at a time. Hence, the framework may face problems dealing with real concurrent actions (Crosby, Jonsson, and Rovatsos 2014), especially when they are *interacting* (Shekhar and Brafman 2020). As a result, the framework is sound and complete only for the problem specifications in which action concurrency is handled carefully. Later, one can find a sound *parallelization* of the agent's plans (by respecting the flexibility proposed for the human agent) in the post-processing phase, e.g., by managing the causal links and synchronizing agents' action orderings. Note that it is not trivial to formally handle interacting actions in the current HATP/EHDA framework, and out of the scope of this work, and it is a part of our future work.

First, the framework builds the whole search space considering all possible, feasible decompositions (Buisan et al. 2022). The framework uses specific actions during the search. *IDLE* is inserted into the agent's plan when its task network is empty or fully decomposed, and *WAIT*, when none of its actions is applicable. Then, it can adapt off-the-shelf graph search algorithms, e.g., the well-known algorithms like A^* and AO^*, and consider social cost, plan legibility, acceptability (Alili, Alami, and Montreuil 2009), etc., to search for a *joint solution plan* for the agents.

A joint solution plan generated, prior to the preprocessing stage, is defined as follows (extending Definition 2 for the HATP/EHDA case).

Definition 3. ***(Joint Solution Plan**.) It is a solution for $\mathcal{P}_{hr}$, represented as a tree or a graph, i.e., $G = (V, E)$. Each vertex ($v \in V$) represents* the ground truth *starting from the root node (i.e., s_0^r). Each edge ($e \in E$) represents a primitive action performed either by the robot (o^r) or the human (o^h). G is branched on possible human choices (o_1^h, o_2^h, ..., o_m^h).*

The requirement from this solution tree is that for each branch, from the root to a leaf node, the sequence of primitive actions, say, $\pi = (o_1^r, o_2^h, o_3^r, ..., o_{k-1}^h, o_k^r)$ is a solution plan for $\mathcal{P}_{hr}$. Here, each o_i^h represents a choice (often out of several choices) the human could make during execution.

Situation Assessment

In human psychology, especially in human-human interaction, spatial reasoning and perspective-taking are crucial. In practice, we often build relations between entities in the environment and estimate the knowledge and capabilities of agents around us to have a seamless collaboration with them.

This work abstractly employs the main ideas of situation assessment based on spatial reasoning and observability during planning. The goal is to predict the situation assessment of a collaborative human agent to have a better estimate of their knowledge and next possible actions, and thus, produce better plans.

Execution-Time Observability Conventions

We formally model run-time observability conventions, understood during plan execution, and use them for task planning in human-robot collaboration.

At a high-level, the formalization below is based on the standard state-variable representation (we refer the reader to (Ghallab, Nau, and Traverso 2004)) such that we abuse the standard notations slightly, sometimes to maintain the flow of the discussion.

A General Understanding of Common Ground

To describe an environment a set of *constant symbols* is used. Constants are often classified as different groups (gr), e.g., *places*, *pots*, *agents*, *etc.* Each constant can be represented as an *object symbol*, e.g., *robot1*, *human2*, *pasta-pkt1*, etc. An *object variable* belonging to a group can be *instantiated* using any constant symbol of that group.

Suppose $\mathcal{S}$ is the complete state-space and $s \in \mathcal{S}$ represents a single state (or a *belief* state in the current context).

Definition 4. ***(State Variable Function.)** It is represented as: $f_{svs} : (?g_1(gr_1), ?g_2(gr_2), ..., ?g_k(gr_k), \mathcal{S}) \rightarrow ?g_{k+1}(gr_{k+1})$.*

Here, svs is *state-variable symbol* (or the attribute name), while each *term* with "?" (a question mark symbol), can be instantiated with a constant symbol from its respective *group* (mentioned inside "()") appears right besides the term. For example, to model agents' location we use $f_{AgtAt} : (?a(Agents), \mathcal{S}) \rightarrow ?r(Rooms)$, representing, for each given legal state ($s_i \in \mathcal{S}$), the room where each agent is located. Each such possible *instantiation* of defined state variable functions represents s_i partially, and is also called a characteristic attribute of s_i.

If a variable is of the form $f_{svs_i} : (?g1(gr_1), \mathcal{S}) \rightarrow ?g_2(gr_2)$ such that gr_1 contains *only* one element, then, for a given state $s \in \mathcal{S}$, we simplify this expression to, $f^s_{svs_i} \rightarrow ?g_2(gr_2)$.

Suppose $s \in \mathcal{S}$ is the real state with the ground truth. As per our assumptions, the belief state of the robot w.r.t. the real-world is always the real world state, i.e., $B^s_{\varphi_r} = s$, and the human belief w.r.t. s is $B^s_{\varphi_h}$ — that is estimated when the robot takes the human's perspective. Each state variable function instantiation w.r.t. a belief state, $B^s_{\varphi_r}$ ($B^s_{\varphi_h}$), represents the truth value of that state attribute (grounded state-variable function) with the *perspective* of φ_r (φ_h). While both these perspectives are managed by the robot.

Definition 5. ***(Belief Divergence.)** Suppose that $f_{svs_i} : (g_{(1)l}, g_{(2)m}, ..., g_{(k)n}, s) \rightarrow g_{(k+1)p}$, for a legal state s and $g_{(i)}$ is the i^{th} group while $g_{(i)j}$ is its j^{th} element, represents a possible* grounding *for a state-variable function. Belief divergence is caused if there exists an instantiation such that $f_{svs_i} : (g_{(1)l}, g_{(2)m}, ..., g_{(k)n}, B_{\varphi_r}) = g_{(k+1)p} \neq f_{svs_i} : (g_{(1)l}, g_{(2)m}, ..., g_{(k)n}, B_{\varphi_h})$.*

Uncertainty in agents' knowledge is not captured explicitly, implying that they are convinced about their beliefs.

Place-Based Observability

We now define place-based observability criteria *inspired by* some standard execution time observability conventions that are relevant for the human-robot context.

Definition 6. ***(Places.)*** *Represented as $Places$, it captures a group of constant symbols such that each member individually captures a pre-specified area in an environment declared by the domain modeler.*

Agents are always situated in a place or moving between two places. Two agents situated in a same place $p \in Places$, at a given time instance, are said to be *co-present*.

Definition 7. ***(Place Specific State Attribute Function.)*** *A* grounded *attribute of a given state is explicitly associated with some* place *or, in a generalized way, it can be expressed as,* $f_{loc} : (f_{svs_i}(?g_1(gr_1), ..., ?g_k(gr_k), \mathcal{S})) \rightarrow Places$

Here, *loc* is a location specific symbol for *place specific attribute*, while the function captures the associated place w.r.t. a state attribute, $f_{svs_i}(...)$.

Defining it this way captures and maintains the explicit places *linked* with "only" the state attributes of our interests. These explicit mappings suggest that an attribute is effective (or *observable*) in its dedicate *place*. Such mappings can change when an action changes the *place* of effectiveness of a state attribute, e.g., *holding* a cup and *moving* to an adjacent room. Such updates are currently manually handled.

Definition 8. ***(State Variable Observability Function.)*** *The state variable observability function maps each grounded state attribute to either* `OBS` *or* `INF`*; or a more general and formal way to express it is,* $f_{obs} : (f_{svs_i}(?g_1(gr_1), ..., ?g_k(gr_k), \mathcal{S})) \rightarrow \{\texttt{INF}, \texttt{OBS}\}$.

Here, *obs* represents an *observability* symbol to capture *state-variable observability* such that a state-variable function, for a legal state $s \in \mathcal{S}$, is classified as either *observable* (`OBS`) or *inferrable* (`INF`). Here, we further generalize it, assuming that an attribute is either *observable* or *inferrable*, and hence, relaxing the individual state based restrictions, that means, the above expression can be further simplified to, $f_{obs} : (f_{svs_i}(?g_1(gr_1), ..., ?g_k(gr_k))) \rightarrow \{\texttt{INF}, \texttt{OBS}\}$.

When a state-variable function ($f_{svs_i}(...)$) belongs to the class `OBS`, it indicates that an agent can assess/observe the correct knowledge of its *exact* status in a given state s_l, if the agent fulfills certain requirements w.r.t. the state s_l. But when a state-variable function ($f_{svs_j}(...)$) belongs to the class `INF`, unlike the previous case, it means that an agent *cannot* observe it directly. However, under certain scenarios, the agent can definitely predict or infer its *exact* status in s_l.

To better understand observability classes, consider the following scenario with a cup of coffee and some sugar on the table (Figure 1). Initially, the cup is far from the man, with no sugar. When he is not attentive, the robot adds some sugar to the cup and pushes it toward him. Once he is attentive again, he would notice that the cup is closer as its position is *observable* to him. But, without seeing the robot adding sugar, he cannot assess if there is sugar in the cup. Hence, we say the *SugarInCup* attribute is *inferrable*, and the human agent would know if they observed the robot act.

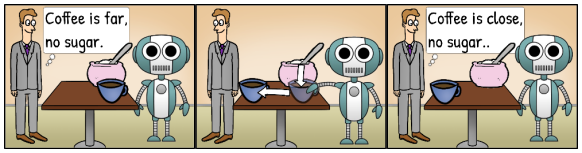

Figure 1: Human is not attentive when Robot adds sugar to the cup before pushing it forward. *Assessing* the new situation after the two actions, Human can only "know" the cup's new position (*observable*), but not *SugarInCup* (*inferrable*).

Belief Updating

An agent's belief can get updated in four ways as follows.

When Acting If the agent executes an action, its belief is updated based on its effects. The state attributes appearing in the action's effect get updated with the new values, while the remaining attributes keep their same old values.

When Being an Observant **(*Action Observability.*)** An action executed by an agent is *observed* by another agent *co-present* throughout the action execution. Therefore, we state that if an agent observes an action getting executed, the agent will infer the *inferrable* effects of the action. Consequently, the agent immediately updates its belief state while each *inferrable* attribute affected by the action receives a new value.

However, when the human executes an action not observed by the robot, we keep it simple and consider that humans make only deterministic moves. The robot does not need to be co-present to get the effects of human actions. And, its belief state is always updated with both their *inferrable* and *observable* effects. More complex cases are part of our future work where the robot's beliefs can diverge, too.

Via Situation Assessment The agent assesses the situation from its current location via spatial reasoning and its reference frame, i.e., the *observability* model. Consequently, it updates its existing belief with the relevant ground truth.

(Based on Definition 7) We can always associate specific state attributes to *places*. For example, suppose that being in the state s_1, the robot switches on the furnace placed in *kitchen*, and this generates a new state s_2. Also, assume that f_{TurnOn} is `OBS` (Definition 8). Then, the place specific attribute function, $f_{loc} : (...)$, w.r.t. the states s_1 and s_2 can be expressed as, $f_{loc}(f^{s_1}_{TurnOn})$ and $f_{loc}(f^{s_2}_{TurnOn})$, respectively, and both map to *kitchen* $\in Places$.

Consider the following scenario: The search progresses from s_2, along $s_1, s_2, ...$, such that the next action applicable in it is, the agent (φ_h) *moving* to the kitchen. This generates a new state s_3, and hence $f_{loc}(f_{AgtAt}(\varphi_h, s_3))$ maps to $kitchen$, but so does $f_{loc}(f^{s_3}_{TurnOn})$. In such cases, φ_h assesses the *status* of the furnace, i.e., the exact value of $f^{s_3}_{TurnOn}$ in s_3, which is `ON`, and hence the human agent updates their belief.

Note that the robot is always aware of the ground reality hence, technically, the idea is effective only for the human agent. Here, the robot takes the human's perspective and performs spatial reasoning as per the human's frame of reference or their current location in the environment. The human's belief is updated w.r.t. what they can see as ground

truth (mimicked by the robot). This enables φ_h to learn an *observable* attribute's value achieved earlier by the robot.

The agent updates its belief with relevant ground truth learned via situation assessment, performed "immediately" (*i.e.*, takes 0 seconds) at each stage, before the execution of the next action. Technically, the robot performs situation assessment for the human and itself.

When Being Communicated The agent's belief is updated by learning the ground truth when another agent communicates an attribute-value pair. Communication occurs via a *distinct set of actions*, modeled as a predefined communication protocol for a pair of sender-receiver agents.

Planning With Communication Actions

For an agent to decide *if*, *when*, and *what* to communicate to another agent, it first needs to establish a common protocol with that agent even before planning. In this work, we establish such communication protocols between each sender-receiver agents pair, where we use speech modality, and model explicit communication actions between two agents. However, we note that these actions are different from agents' (non-) primitive actions, and they are used only in special circumstances during planning and execution.

Modeling Communication Actions

Next, we propose a generic communication action schema (ca) in this context. Suppose there exists a state-variable function, $f_{svs} : (?g_1(gr_1), ?g_2(gr_2), ..., ?g_k(gr_k), \mathcal{S}) \rightarrow ?g_{k+1}(gr_{k+1})$, such that for the current world state $s \in \mathcal{S}$, $f_{svs}(g_{(1)l}, g_{(2)m}, ..., g_{(k)n}, s)$ maps to q.

The agent φ_i can *communicate* this attribute-value pair to φ_j (*via* the action $ca_{\varphi_i,\varphi_j}(f_{svs}(...), q)$), if the following conditions (i.e., its preconditions) meet: (1) For φ_i, $f_{svs}(g_{(1)l}, g_{(2)m}, ..., g_{(k)n}, B^s_{\varphi_i}) = q$, and (2) for φ_j, $f_{svs}(g_{(1)l}, g_{(2)m}, ..., g_{(k)n}, B^s_{\varphi_j}) = r$ s.t. $r \neq q$. Later, φ_j's belief is updated, i.e., $f_{svs}(g_{(1)l}, g_{(2)m}, ..., g_{(k)n}, B^s_{\varphi_j}) \leftarrow q$.

Planning with Reasoning on Human Mental State

Now that we have defined and described all essential tools, we describe the new approach and explain how these tools are utilized by it. First, we note that we put an effort to make our main contributions to have a "generic" state-variable representation. Hence, we believe that they are not limited to only our intended architecture (HATP/EHDA) in this paper. We plan to substantiate this claim experimentally in future.

We only describe the *main changes* made to enhance the underlying solver. Suppose an agent applies an action: First, the belief states of all agents co-present with this agent get updated with the action's inferrable effect. Later, the *situation assessment* process is used for every other agent to assess the changes (captured via observable state attributes). As a result, their beliefs are updated by the observable effects if the required conditions are satisfied. We discussed these subroutines in detail in earlier sections.

Communication in Planning The following essential changes in the existing algorithm support efficient planning with communication. (Recall the sender-receiver pair discussed earlier.) The approach identifies whether the receiver's belief divergence is *relevant*, and thus if the receiver's belief needs to be corrected. If so, all state attributes where the receiver agent's belief differs w.r.t. the ground truth are identified. Then, we decide which of these state attributes and their exact values must be transmitted by the sender (*i.e.*, the robot), supporting the idea of communicating minimally and avoiding verbosity. Once we determine that, the receiver's (*i.e.*, human's) belief gets updated accordingly, ensuring that the remaining false knowledge of the receiver's updated belief is *not effective*, which also means that the belief divergence is not relevant anymore.

- **Relevant Belief Divergence.** At a given stage, if the human agent, based on their own belief, can perform a set of actions that differs from the one the human could perform w.r.t. to the robot's belief (or the ground reality), then we say that such divergence in their belief is *relevant*. A divergence is also declared as *relevant* if an action has different effects w.r.t the other agent's belief. However, currently, our reasoning system is myopic: We do not evaluate in a principled way the impact of *different* actions that humans could execute (based on their wrong belief), or of *different* effects, with their overall positive and detrimental effects on achieving the joint task. At this stage, it simply "decides" to align the agent's beliefs using communication actions whenever the belief divergence is relevant. However, a smarter approach would probably be to analyze the history of plan trace(s) or agents' future actions to decide whether their beliefs should get aligned or not, or for that matter, how much to communicate, but it is currently out of the scope of this work.
- **Communicate Only The Required Facts.** It "decides" the *key* changes in the human's belief to be made such that the relevance of their updated belief (might still be diverging) is non-effective. The subroutine that decides needed communication to be made appears in the robot's executable policy; is described in the following steps:

  1. *Store* each attribute and its value if the attribute's value differs in the human's belief from the robot's belief.
  2. For each stored attribute-value pair, *build* a communication action $ca_{\varphi_r,\varphi_h}(...)$ based on the schema described earlier. They are considered equally costly.
  3. At a given planning stage, *e.g.*, the ground truth s_i, follow the Breadth-First Search ordering. The *source* is $B^{s_i}_{\varphi_h}$, and each $ca_{\varphi_r,\varphi_h}(...)$ changes and aligns *exactly* one attribute while its updated value satisfies the ground truth. Applying $ca_{\varphi_r,\varphi_h}(...)$ generates a new belief state following state transition rules, i.e., $B^{s_i,1}_{\varphi_h} = \gamma(B^{s_i}_{\varphi_h}, ca_{\varphi_r,\varphi_h}(...))$. It continues until the first (updated) belief is *selected* to expand s.t. its remaining divergence is ineffective. The actions used from the root until the current belief state are *retrieved*.

  Once the above subroutine finishes, the retrieved action set $CA = \{ca^i_{\varphi_r,\varphi_h}(...)\}$ is utilized for belief alignment.

Agents may begin with different beliefs. Situation assessment (SA) can update human belief state, but humans cannot know inferable facts. So, we redefine Definition 3 to make

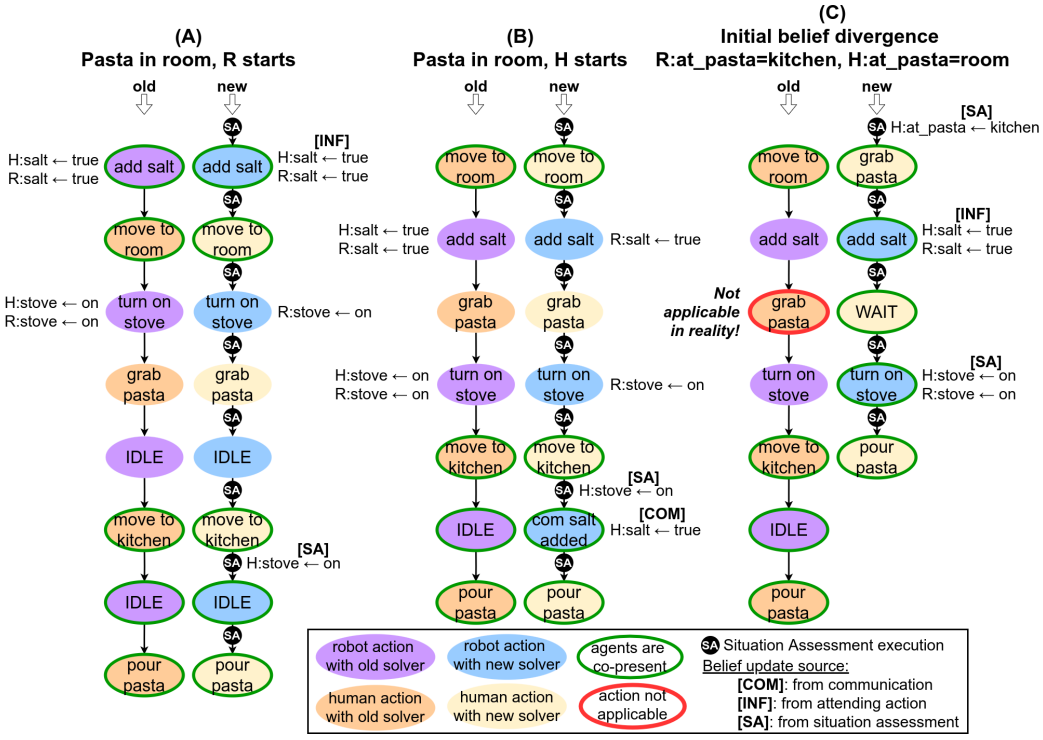

Figure 2: Shows the plans obtained in three scenarios: Each scenario presents two plans — on the left, obtained by the old solver, and on the right, obtained via the proposed solver. The latter depicts more realistic and appropriate belief updates focusing on two attributes. Case (A): the plans are the same, but the updates in the human belief are more realistic with our approach. Case (B): the human has no certainty on *SaltInPot*, while ours decides to communicate to remove the ambiguity. And, Case (C): the initial belief divergence induces an "invalid" plan, but our approach predicts that the human agent will *assess the situation* and update their belief with all the facts without being communicated. (Other minor details are outlined in the figure.)

it sound w.r.t. our new planning approach. Assume at each step ($t = 0, 1, 2, ...$), humans perform SA, while the robot executes each communication action $ca \in CA$, such that the human's belief state *updates immediately* (takes 0 seconds). Later, following the updated beliefs, the human's or robot's regular primitive action is executed, and the effects only appear in the next step. Hence, the other two types of belief updates happen post the action execution. And, continues.

Empirical Evaluation

Before we describe the two domains used for the experiments and the results to compare the old and new approaches, we first make a *high-level* distinction between the existing and new approaches. In principle, the existing solver can handle agents' individual belief divergences, but "not during planning" unlike our approach. To achieve that, the existing solver can use a cumbersome technique that intervenes by updating the (collaborative) task networks of the joint task model/specifications while using triggers.

Cooking Pasta Domain Suppose a stove and salt are available in *Kitchen* (*Places*), and the pasta is either in *Kitchen* or *Room* – they are adjacent. The agents have different roles and can only operate in the two places. Robot *adds* salt to the pot and *turns-on* the stove. Human *grabs* the pasta and *pours* it into the pot. The human can add pasta, but only after salt gets added to the pot and the stove is ON.

Focus on the following two attributes: For a given state, $s_i \in \mathcal{S}$, $f^{s_i}_{SaltInPot} \in \{true, false\}$ and $f^{s_i}_{stove} \in \{on, off\}$ such that only $f^{s_i}_{SaltInPot}$ is *inferrable*, all others belong to OBS.

Preparing Box Domain A box filled with a fixed number of balls and with a sticker pasted on is considered prepared and needs to be sent. Both the agents can *fill* the box with balls from a bucket, while only the robot can *paste* a sticker, and only the human can *send* the box. The bucket can run out of balls, so when there is only one ball left, the human *moves* to another room to *grab* more balls and *refill* it. A sticker pasted on the box is *observable*, while the number of balls in the box is *inferrable*. Other attributes belong to OBS.

Experiments

Qualitative analysis in the first domain mentions *subtleties* the old solver overlooks and how ours is aware of them, updates belief, and effectively manages divergences. Then, quantitative studies compare the solvers in both domains.

Qualitative Analysis Consider the first domain. We discuss the plans obtained in three different scenarios as shown in Figure 2. Assume that both the agents are in *Kitchen* and the pasta is in *Room*. Scenario (A) captures the agents' plans

Domain	*Old Solver*			*Our Solver*	
	S	*NA*	*IDL*	*S*	*Com*
Cooking	18.6%	77.0%	23.0%	100%	54.9%
Box	25.0%	83.3%	16.7%	100%	68.8%
Average	21.8%	80.2%	19.9%	100%	61.9%

Table 1: In each domain: for the *old solver*, the success rate (*S*), the ratio of failed plans due to a non-applicable action (*NA*), and the ratio of failed plans due to an inactivity deadlock case (*IDL*), while for *our solver*, the success rate (*S*) and the ratio of plans including a communication action (*Com*).

when the robot starts while Scenario (B) shows when the human starts. Let us change the scenario: Suppose, in hindsight, the pasta is moved to *kitchen* such that the robot knows it, while the human still has the old belief. For this, Scenario (C) captures the plans obtained when the human starts.

- **Scenario (A):** Although the solvers generate similar plans, they update the human's belief differently if and when the human and robot are co-present. E.g., turning on the stove ideally (realistically) does not affect the human's mental state, which is not the case for the old solver. It considers agents omniscient, so the human knows everything immediately once achieved. Our solver predicts it when the human returns to the kitchen and assesses that the stove is ON, and their belief gets updated.
- **Scenario (B):** Human leaves the kitchen. Practically, they are unaware of the changes achieved in the environment by the robot's actions: *turn-on* & *add-salt*. The old solver generated a similar plan as in *Sce. (A)* where the human already knew that the stove was ON, and salt was added. However, an appropriate situation assessment and inference based on action observability together guarantee that, when returning to the kitchen, the human assesses that the stove is ON. Moreover, the robot predicts that the human cannot know the *SaltInPot* fact and that it is a relevant divergence that needs to be handled via communication.
- **Scenario (C):** With the old solver, the human moves to *Room* to *grab* the pasta, but in reality, being an illegal action w.r.t. the robot's belief or the ground truth, it *fails*. In our case, the human agent assesses the environment to update their belief state, knowing the pasta is in the kitchen. Hence, no communication is required.

Quantitative Studies The HATP/EHDA framework's current solver and ours are tested and compared on two domains in Table 1. We consider (in box domain) *three* boxes to be prepared and sent. While in both, we generated 512 different initial states, including 448 (87.5%) with divergent initial agents' beliefs and 64 states where both agent beliefs are fully aligned initially. Overall, 2048 plans were generated.

With the old approach, a planning failure occurs due to: (a) an action of a plan not applicable in another agent's belief state, including the ground truth; (b) if an inactivity deadlock occurs, which is assumed to be the case after a succession of at least four *WAIT* and (or) *IDLE* actions. A deadlock occurs when the human has a belief divergence and waits for a never-happening robot's action, e.g., waiting for *add salt*, but *SaltInPot* is already achieved. For the old solver, the success rate (*S*), the *ratios* of the number of failed plans due to a inapplicable action (*NA*) and an inactivity deadlock (*IDL*) appear in the table. For ours, the success rate (*S*) is shown and the *ratio* of successful plans including a communication action is presented under *Com*.

As the existing solver does not handle belief divergence in planning, the applicability of actions was never an issue w.r.t. another agent's belief. Therefore, if the *IDL* case occurs, it is understood that the task specification is erroneous w.r.t. the old problem specification.

Our solver always finds legal plans, and on average, *approx.* 62% of them use *communication*. Thus, the robot doesn't need to communicate systematically as assessing situations handles a major part of the divergences (87.5% of the scenarios have divergent beliefs initially). For the old solver, if no initial belief divergence exists, it always finds a legal plan, considering the agents omniscient. E.g., Fig. 2(A). However, sometimes, this causes problems in practice. Scenarios beginning with distinct beliefs induce actions often not applicable in another agent's belief state (or the ground truth), evident by the (average) 21.8% success rate.

Discussion

Formalizing run-time observability conventions is crucial for planning as ignoring them may lead to problems in practice. E.g., Fig. 2(B) showing human knowing *SaltInPot*, is ambiguous in reality. We describe a far more realistic way to estimate the evolution of human belief during planning, using the situation assessment discussed in *Qualitative Analysis*. Explicit reasoning on the human mental state detects and prevents ambiguous situations with communication while also prohibiting the robot from being too verbose, as shown in *Quantitative Studies*. Compared to the old process, this produces *consistent* and more *robust* plans overall.

However, our approach does not *refute* something believed by an agent through situation assessment without assessing its exact true value. E.g., for some $s_i \in \mathcal{S}$, if the human wrongly believes that the pasta is in *Kitchen*. The situation assessment does not help refute this, while the human is in *Kitchen*. The reason is that $f^{s_i}_{PastaNotInKitchen}(...)$ is not modeled explicitly as an attribute. Such issues do not affect the solver's *completeness* as far as the situation assessment is concerned. Moreover, our solver handles them as a relevant divergence to be aligned. Thus, the human is just communicated with correct updates.

Summary

Building on earlier work, we presented a new method to model execution-time observability conventions appropriate for HRI and to use them to estimate the evolution of the human mental state. It is based abstractly on situation assessment and action observability criteria. A new planning approach is described, utilizing this better estimation of human mental state to plan for more robust and consistent human-robot joint activities such that a relevant belief divergence is tackled by explicitly modeled communication actions.

References

Alami, R.; Chatila, R.; Fleury, S.; Ghallab, M.; and Ingrand, F. 1998. An Architecture for Autonomy. *Int. J. Robotics Res.*, 17(4): 315–337.

Alami, R.; Clodic, A.; Montreuil, V.; Sisbot, E. A.; and Chatila, R. 2006. Toward Human-Aware Robot Task Planning. In *AAAI spring symposium: to boldly go where no human-robot team has gone before*, 39–46.

Alili, S.; Alami, R.; and Montreuil, V. 2009. A task planner for an autonomous social robot. In *Distributed autonomous robotic systems 8*, 335–344. Springer.

Alili, S.; Warnier, M.; Ali, M.; and Alami, R. 2009. Planning and plan-execution for human-robot cooperative task achievement. In *19th international conference on automated planning and scheduling (ICAPS)*.

Annett, J.; and Duncan, K. D. 1967. Task analysis and training design. *Journal of Occupational Psychology*.

Baron-Cohen, S.; Leslie, A. M.; and Frith, U. 1985. Does the autistic child have a "theory of mind"? *Cognition*, 21(1): 37–46.

Berlin, M.; Gray, J.; Thomaz, A. L.; and Breazeal, C. 2006. Perspective taking: An organizing principle for learning in human-robot interaction. In *AAAI*, volume 2, 1444–1450.

Buckingham, D.; Chita-Tegmark, M.; and Scheutz, M. 2020. Robot Planning with Mental Models of Co-present Humans. In *International Conference on Social Robotics*, 566–577. Springer.

Buisan, G. 2021. *Planning For Both Robot and Human: Anticipating and Accompanying Human Decisions.* Ph.D. thesis, INSA Toulouse, France.

Buisan, G.; and Alami, R. 2021. A Human-Aware Task Planner Explicitly Reasoning About Human and Robot Decision, Action and Reaction. In Bethel, C. L.; Paiva, A.; Broadbent, E.; Feil-Seifer, D.; and Szafir, D., eds., *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2021, Boulder, CO, USA, March 8-11, 2021*, 544–548. ACM.

Buisan, G.; Favier, A.; Mayima, A.; and Alami, R. 2022. HATP/EHDA: A Robot Task Planner Anticipating and Eliciting Human Decisions and Actions. In *IEEE International Conference On Robotics and Automation (ICRA 2022)*. Philadelphia, United States.

Buisan, G.; Sarthou, G.; and Alami, R. 2020. Human Aware Task Planning Using Verbal Communication Feasibility and Costs. In Wagner, A. R.; Feil-Seifer, D.; Haring, K. S.; Rossi, S.; Williams, T. E.; He, H.; and Ge, S. S., eds., *Social Robotics - 12th International Conference, ICSR 2020, Golden, CO, USA, November 14-18, 2020, Proceedings*, volume 12483 of *Lecture Notes in Computer Science*, 554–565. Springer.

Crosby, M.; Jonsson, A.; and Rovatsos, M. 2014. A Single-Agent Approach to Multiagent Planning. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, 237–242. IOS Press.

De Silva, L.; Lallement, R.; and Alami, R. 2015. The HATP hierarchical planner: Formalisation and an initial study of its usability and practicality. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 6465–6472. IEEE.

Devin, S.; and Alami, R. 2016. An implemented theory of mind to improve human-robot shared plans execution. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 319–326. IEEE.

Flavell, J. H. 1992. Perspectives on perspective taking. *Piaget's theory: Prospects and possibilities*, 107–139.

Ghallab, M.; Nau, D. S.; and Traverso, P. 2004. *Automated planning - theory and practice*. Elsevier. ISBN 978-1-55860-856-6.

Hoffman, G.; and Breazeal, C. 2007. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 1–8.

Ingrand, F. F.; Chatila, R.; Alami, R.; and Robert, F. 1996. PRS: A high level supervision and control language for autonomous mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, 43–49. IEEE.

Johnson, M.; and Demiris, Y. 2005. Perceptual perspective taking and action recognition. *International Journal of Advanced Robotic Systems*, 2(4): 32.

Lallement, R.; De Silva, L.; and Alami, R. 2014. HATP: An HTN planner for robotics. *arXiv preprint arXiv:1405.5345*.

Lallement, R.; de Silva, L.; and Alami, R. 2018. HATP: hierarchical agent-based task planner. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*.

Lemaignan, S.; Warnier, M.; Sisbot, E. A.; Clodic, A.; and Alami, R. 2017. Artificial Cognition for Social Human-Robot Interaction: An Implementation. *Artificial Intelligence*, 247: 45–69.

Martinie, C.; Palanque, P.; Bouzekri, E.; Cockburn, A.; Canny, A.; and Barboni, E. 2019. Analysing and demonstrating tool-supported customizable task notations. *Proceedings of the ACM on human-computer interaction*, 3(EICS): 1–26.

Milliez, G.; Lallement, R.; Fiore, M.; and Alami, R. 2016. Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 43–50. IEEE.

Milliez, G.; Warnier, M.; Clodic, A.; and Alami, R. 2014. A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management. In *The 23rd IEEE international symposium on robot and human interactive communication*, 1103–1109. IEEE.

Nikolaidis, S.; Kwon, M.; Forlizzi, J.; and Srinivasa, S. 2018. Planning with verbal communication for human-robot collaboration. *ACM Transactions on Human-Robot Interaction (THRI)*, 7(3): 1–21.

Paternò, F. 2004. ConcurTaskTrees: an engineered notation for task models. *The handbook of task analysis for human-computer interaction*, 483–503.

Premack, D.; and Woodruff, G. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4): 515–526.

Roncone, A.; Mangin, O.; and Scassellati, B. 2017. Transparent role assignment and task allocation in human robot collaboration. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1014–1021. IEEE.

Sanelli, V.; Cashmore, M.; Magazzeni, D.; and Iocchi, L. 2017. Short-term human-robot interaction through conditional planning and execution. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

Sebastiani, E.; Lallement, R.; Alami, R.; and Iocchi, L. 2017. Dealing with on-line human-robot negotiations in hierarchical agent-based task planner. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

Shekhar, S.; and Brafman, R. I. 2020. Representing and planning with interacting actions and privacy. *Artif. Intell.*, 278.

Sisbot, E. A.; Ros, R.; and Alami, R. 2011. Situation assessment for human-robot interactive object manipulation. *2011 RO-MAN*, 15–20.

Tellex, S.; Knepper, R. A.; Li, A.; Rus, D.; and Roy, N. 2014. Asking for Help Using Inverse Semantics. In Fox, D.; Kavraki, L. E.; and Kurniawati, H., eds., *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*.

Trafton, J. G.; Cassimatis, N. L.; Bugajska, M. D.; Brock, D. P.; Mintz, F. E.; and Schultz, A. C. 2005. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 35(4): 460–470.

Tversky, B.; Lee, P.; and Mainwaring, S. 1999. Why do speakers mix perspectives? *Spatial cognition and computation*, 1(4): 399–412.

Unhelkar, V. V.; Li, S.; and Shah, J. A. 2019. Semi-Supervised Learning of Decision-Making Models for Human-Robot Collaboration. In Kaelbling, L. P.; Kragic, D.; and Sugiura, K., eds., *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, 192–203. PMLR.

Unhelkar, V. V.; Li, S.; and Shah, J. A. 2020a. Decision-making for bidirectional communication in sequential human-robot collaborative tasks. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 329–341.

Unhelkar, V. V.; Li, S.; and Shah, J. A. 2020b. Decision-Making for Bidirectional Communication in Sequential Human-Robot Collaborative Tasks. In Belpaeme, T.; Young, J. E.; Gunes, H.; and Riek, L. D., eds., *HRI '20: ACM/IEEE International Conference on Human-Robot Interaction, Cambridge, United Kingdom, March 23-26, 2020*, 329–341. ACM.

Warnier, M.; Guitton, J.; Lemaignan, S.; and Alami, R. 2012. When the robot puts itself in your shoes. Managing and exploiting human and robot beliefs. In *IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 7p. Paris, France.