Incorporating Helpful Behavior into Collaborative Planning

Incorporating Helpful Behavior into Collaborative Planning

Ece Kamar, Ya'akov Gal and Barbara J. Grosz
SEAS, Harvard University
Cambridge, MA 02138
{kamar, gal, grosz} @eecs.harvard.edu

ABSTRACT

This paper considers the design of agent strategies for deciding whether to help other members of a group with whom an agent is engaged in a collaborative activity. Three characteristics of collaborative planning must be addressed by these decision-making strategies: agents may have only partial information about their partners' plans for sub-tasks of the collaborative activity; the effectiveness of helping may not be known a priori; and, helping actions have some associated cost. The paper proposes a novel probabilistic representation of other agents' beliefs about the recipes selected for their own or for the group activity, given partial information. This representation is compact, and thus makes reasoning about helpful behavior tractable. The paper presents a decision-theoretic mechanism that uses this representation to make decisions about two kinds of helpful actions: communicating information relevant to a partner's plans for some sub-action, and adding domain actions that are helpful to other agent(s) into the collaborative plan. This mechanism includes a set of rules for reasoning about the utility of helpful actions and the cost incurred by doing them. It was tested using a multi-agent test-bed with configurations that varied agents' uncertainty about the world, their uncertainty about each others' capabilities or resources, and the cost of helpful behavior. In all cases, agents using the decision-theoretic mechanism to decide whether to help outperformed agents using purely axiomatic rules.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Collaborative computing*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Experimentation

Keywords

Collaboration, Multi-agent Decision Making, Helpful Behavior, Communication, SharedPlans

Cite as: Incorporating Helpful Behavior into Collaborative Planning, Ece Kamar, Ya'akov Gal and Barbara Grosz, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX.

1. INTRODUCTION

Collaboration is a special type of group activity in which agents work together toward a shared goal, typically the performance of a collective action. The participants in a collaborative activity form coordinated, mutually supportive plans. They make commitments to the group activity, to doing some of the constituent tasks of that activity, and to other participants' ability to accomplish other constituent actions [6, 10, 9, 5]. As in most multi-agent task settings, the collaborative activity is carried out in a world that is constantly changing, the participants' knowledge about the world is inherently incomplete, individuals have (sensory) access to different parts of the world, and their beliefs–including their beliefs about how best to perform an action–may differ.

Although the participants in a collaboration have an incentive to help others by the nature of their commitments to the shared goal and to each others' actions in service of satisfying that goal, a decision about whether to help still requires deliberation. Helpful actions result in some cost to the agent helping, and they may incur costs for the group activity as a whole. Usually costs include resources consumed in communicating, lost opportunities to do other activities, and the need for group members to adapt their individual plans to the helpful act or its effects. Thus, even in collaborative settings, agents must weigh the trade-off between the potential benefit to the group of some helpful behavior and its associated costs.

This paper addresses the intertwined problems of recognizing when help is needed in a collaboration and determining whether to help, taking into account the costs of a helpful action and its possible effects on the beliefs and commitments of group members. It is specifically concerned with collaborative activities that take place in settings in which there is uncertainty about agents' capabilities and about the state of the world.

Throughout the paper, we will illustrate various aspects of this helpful-behavior decision-making problem using an example of two agents, Alice and Bob, who are cooking for a dinner party. Alice intends to make the entree and Bob intends to make an appetizer. Following the SharedPlans formalization of collaborative activity [6], we assume that only the agents performing some subactivity know the full details of how they are doing that activity. Thus, Alice may not know what appetizer Bob is preparing, and Bob may have limited information about Alice's entree. Suppose Alice believes that Bob may be preparing stuffed mushrooms or some type of salad, and she discovers that one of the guests is allergic to mushrooms. Alice needs to decide whether to communicate this information to Bob. Her entree recipe may be ruined if she takes too long to find Bob and inform him. She needs to reason not only about this cost, but also about the likelihood that Bob's appetizer recipe involves mushrooms.

We consider two distinct types of helpful behavior: performing a helpful action that is outside the scope of an agent's individual responsibility in the collaborative activity (e.g., Alice's buying an ingredient for Bob's appetizer); and, communicative actions, including two sub-types: informing actions (e.g., Alice's telling Bob that one of the guests is allergic to mushrooms) and asking actions (e.g., Alice's inquiring of Bob whether any of the dinner guests are vegetarian). We provide a general decision-theoretic mechanism for reasoning about the utility of performing each of these behaviors in the context of agents' commitment to the group activity. This mechanism was evaluated empirically using the Colored Trails platform [7] configured to represent situations in which individual agents perform different tasks, group success depends on joint performance, and the world is dynamic. The experiments varied the cost of helpful behavior as well as agents' uncertainty about the world and about the capabilities of other agents with which they collaborated. In all cases, agents performed better using this mechanism to make helpful-behavior decisions than using purely axiomatic methods.

The paper makes three major contributions. First, it expands the SharedPlan formalization by defining a set of rules that formalize the way that helpful behavior arises from the commitments and intentions of the participants in a collaborative activity. This extension formally integrates decision theoretic and BDI models. It thus fills a gap in teamwork theories by incorporating costs and uncertainty into a BDI model (SharedPlans) in a principled and general way, and by making decision theoretic reasoning tractable despite incomplete information. Second, it defines a decision-theoretic mechanism that deploys these rules to refine agents' plans, adding helpful behavior actions when appropriate. Third, it defines a novel compact representation, Probabilistic Recipe Trees (PRT), which is used by the mechanism to accommodate differences among agents' beliefs about the possible recipes that other agents are using. Without this compact representation of uncertainty, integration of BDI and decision-theoretic models would require intractable computation of a complete policy for all of the agents in a collaboration.

In Section 3, we present basic definitions for constructs used in our model. Section 4 presents a novel probabilistic representation of the possible recipes selected by a group. Section 5 defines a set of rules for engaging in helpful behavior that rely on this representation. Section 6 describes the empirical evaluation of the decision-theoretic model.

2. RELATED WORK

Several formalizations have been proposed to model collaboration and teamwork [8, 6, 1, 10, 2, 9], all of which recognize communication as a major requirement for successful and cohesive collaborative activity. The research described in this paper differs in considering communication as a special form of helpful behavior and then examining the general question of when to help. In addition, it provides a decision-theoretic rather than axiomatic approach to modeling helpful-behavior decisions.

Several prior approaches have axiomatized decisions to communicate or to help in the context of formalizations of collaboration in terms of the intentions, beliefs and mutual beliefs of the participants. Cohen and Levesque's axiomatic approach stipulates that agents communicate to the group whenever a goal is discovered impossible to achieve [1, 10]. Fan et al. extend the set of logical axioms to provide for proactive information exchange (informing and asking for information) [3]. These approaches do not consider the cost of communication nor do they provide mechanisms for helpful domain actions that improve the utility of plans.

The SharedPlan (SP) formalization includes axioms that entail adopting intentions for helpful acts, or lead to communication based on certain kinds of intentions in the SP specification [6]. These axioms represent both the benefit of a helpful action to the group activity and the costs to the individual performing the helpful action. However, they do not handle uncertainty regarding the world or agents' capabilities. Furthermore, the specification provides no insight on how these axioms can be realized or implemented in agent design, whereas this paper provides a decision-making mechanism.

STEAM, which drew on both the joint intensions [1, 10] and the SharedPlans [6] theories, supported the construction of agents able to collaborate in complex, real world domains of military training and robot soccer [13]. It included a decision-theoretic mechanism for communication which modeled the cost-benefit trade-off associated with communicating information to the full group. This mechanism constructed a decision-tree for each agent every time a communication action was considered. This had significant complexity costs for agents that needed to consider many such actions. The mechanism defined in this paper has lower complexity and is more general.

Work on decentralized approaches to multi-agent planning have provided models that consider the cost-benefit tradeoffs of communication among agents [4]. As helpful behavior can emerge between any agents in the collaborative activity, the helpful behavior needs to be directly embedded in the joint policy of the whole group of agents, making it exponential in the size of the history of agents' observations. Refining agents' plans in this setting means updating their entire policy every time a helpful action is considered, which is infeasible.

3. BASIC DEFINITIONS

In this section, we briefly describe key aspects of the SharedPlan formalization of collaborative activity and define other constructs we will be using through the paper. A SharedPlan comprises a set of beliefs about the actions to be performed and intentions of the agents working together in a collaborative activity. The formalization uses the concept of a recipe as a set of sub-actions and constraints such that performing those sub-actions under those constraints constitutes completing the action [11]. The actions in the recipe may be basic-level actions (executable at will) or complex actions that decompose into other complex or basic-level actions.

The SharedPlan formalization deploys two intentional attitudes, intending to (do an action) and intending that (a proposition hold). Intentions-to are used to represent an agent's commitments to its own actions, whether for in-

$Context(C_\alpha, G_1, \alpha, T)$	the believed context of action α
$cba.basic(G_1, \beta, C_\beta)$	probability that can bring about action β
$cost.basic(G_1, G_2, \beta, C_\beta)$	cost of action β
$V(G_1, \alpha, C_\alpha)$	value of action α

Table 1: Summary of predicates and functions

dividual ends or as part of a group activity. Intentions-that are used to represent an agent's commitments toward the group activity and the actions of its partners in service of that activity. Following SharedPlans, we represent agent G_1's commitment to do an action α in context C_α as $Int.To(G_1, \alpha, C_\alpha)$. $Int.Th(G_1, prop, C_{prop})$ represents G_1's intention that proposition *prop* holds in context C_{prop}.

For the purposes of this paper, we define the context in which an action is performed by an agent or a group of agents as the combination of the information that agents use to make decisions about that action at a particular time. For example, Alice's context for meal-making includes Alice's beliefs about the world (e.g., whether there are fresh tomatoes in the house), and Alice's beliefs about Bob's context for meal-making (e.g., Alice's beliefs about Bob's knowledge about whether there are tomatoes).

Table 1 summarizes several new predicates and functions used in this paper. The predicate *Context* is true if C_α is the context in which agent G_1 believes at time T that action α is being done. The function *cba.basic* refers to the probability that agent G_1 can bring about (i.e., successfully complete) a basic-level action β in context C_β. Similarly, *cost.basic* refers to the cost incurred by agent G_1 when basic-level action β is executed by agent G_2 in context C_β (where G_1 and G_2 may refer to the same agent).

The function V represents the non-negative utility value for agent G_1 for successful completion of action α in context C_α. If an action is a basic level action, the utility for performing the action is equal to the V-function value. The utility for carrying out a complex action based on a given recipe includes the value of V for the action, as well as the sum of the utilities for performing each of its sub-actions. Thus, for a complex action, the utility may be more than "the sum of its parts". For example, the utility for making dinner may be higher than the combined utilities of making the individual dishes. The valuation of an action may be zero if accomplishing that alone does not provide any utility. For example, suppose that Alice and Bob fail to make dinner, because Alice did not prepare an entree. However, Bob prepared an appetizer. Their utility will be higher than in the case where neither Alice nor Bob prepared their respective dishes. In contrast, the utility for chopping onions is zero if the plan for mushroom puffs fails.

We assume that *cba.basic*, *cost.basic* and V are intrinsic properties of actions and that agents either know or have an estimate of these values in their mental models. We also assume that agents engaging in a collaboration aim at maximizing some agreed upon and commonly known utility of the group. This utility is a computable value, which is typically some combination of individual agents' utilities and some other costs weighted appropriately. For the purposes of the paper, we assume that agents are truthful, and they do not exhibit malicious behavior.

4. PROBABILISTIC RECIPE TREES

Key features of the SharedPlan formalism are that agents' plans may be partial, agents may have incomplete information about the way to accomplish a group activity, and agents are responsible for different constituent actions. For example, Alice may not know the appetizer Bob is making, but they still have a joint commitment to cook dinner. However, agents cannot reason about the benefit to the group from engaging in a helpful action when they have no information about the recipes that other group members are considering. To bridge this gap, we introduce a novel representation, Probabilistic Recipe Trees (PRTs), which enable agents to represent their beliefs about the recipes that may be selected by group members to complete a collaborative activity.

A Probabilistic Recipe Tree (PRT) for an action α is a structured tree representation that defines a complete probability distribution over the possible recipes for accomplishing α. Each node in a PRT represents an action and has several properties associated with the action (e.g., the set of agents responsible for carrying out the action). Leaf nodes represent basic-level actions, and intermediate nodes represent complex actions. Intermediate nodes may be either AND or OR nodes. Each child of an AND node represents a constituent sub-action of a recipe for completing the AND node action. Each child of an OR node represents a possible choice of a recipe for the OR node action, where the choice is non-deterministic. Each branch from an OR node to one of its children nodes has an associated probability representing the likelihood that the child node is selected as a recipe for the OR node action. Figure 1 presents a PRT for preparing a dinner consisting of an appetizer and an entree. The children of the top node of the tree (making dinner) are making an appetizer and making an entrée. The children of the appetizer node represent possible recipes for an appetizer: mushroom puffs, lettuce salad, and tomato salad. The likelihood of selecting mushroom puffs as the appetizer in this example is 0.7.

Thus, a PRT defines a probability distribution over possible recipes for completing the action associated with the root node. The PRT of Figure 1 includes a probability distribution over 9 possible recipes for making dinner. An assignment of recipes to all of the OR nodes constitutes one deterministic recipe for achieving this action. For example, one possible recipe for meal-making is making mushroom puffs and chicken with tomato sauce. The probability of choosing this recipe is 0.35.

A PRT is exponentially more compact than an exhaustive representation over a set of recipes. That this is the case may be seen by considering the space of recipes with up to n potential recipes for each action, with each recipe having up to m constituent actions, and d representing the number of levels of decomposition needed to transform the top-level action into basic-level actions. The size of an exhaustive tree representation for each possible recipe in this setting is $O(m^d)$. Because there are n possible recipes for each action, the number of possible trees is $O(n^{m^d})$, and a distribution over recipes will have to assign a separate probability for each of them. In contrast, the size of the PRT for this example is $O((nm)^d)$, which is exponentially smaller.

A PRT is also a modular representation. The following three operators may be used to restructure a PRT as agents refine their recipes with helpful actions.

Addition: The operator $PRT_\alpha \bigcup PRT_\beta$ adds PRT_β as a

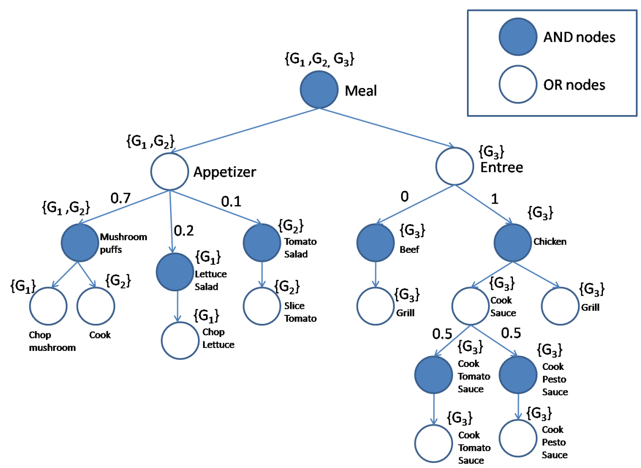

Figure 1: A Probabilistic recipe tree for the meal preparation example. Each node is associated with an action (e.g., make mushroom puffs), and a set of agents to perform the action (e.g., $\{G_1, G_2\}$).

child of PRT_α. If α is an OR node, then the probability distribution over the branches leaving the OR node is normalized.

Subtraction: The operator $PRT_\alpha \setminus PRT_\beta$ removes the sub-tree PRT_β from PRT_α, if such a sub-tree exists. If α is an OR node, then the probability distribution over the branches leaving the OR node is normalized.

Replacement: The operator $PRT_\alpha \bigotimes PRT'_\beta$ removes PRT_β (the original PRT for β in PRT_α) from PRT_α, adds PRT'_β to the parent node of PRT_β. If this parent is an OR node, the probability distribution over the branches leaving the node is normalized. The PRT replacement operation is more than subtraction followed by addition in that it gets from the subtraction the node at which the new subtree will be added (eliminating the need for search).

The decision-theoretic analysis of helpful behavior requires computing the costs and benefits of performing an action based on the selected recipes. To this end, we utilize the following set of functions to represent agents' beliefs for the recipes selected for an action, and to evaluate the costs and utilities of these recipes.

The function $p\text{-}CBA(PRT_\alpha, C_\alpha)$ represents the probability of successfully performing action α in context C_α given the recipes represented in PRT_α. For leaf nodes representing basic actions, the function returns a value that equals the function *cba.basic* applied to the leaf. For AND nodes the function returns the product of the probabilities that the children nodes will succeed. For OR nodes, the function returns an average of the likelihood that the child nodes will succeed, weighted by the probability assigned to the child.

The $Cost(G_i, PRT_\alpha, C_\alpha)$ function represents the expected cost to agent G_i for the group carrying out the recipes represented in PRT_α in context C_α. For leaf nodes, this function returns the value of the function *cost.basic* applied to the leaf. For AND nodes this function is a summation of the cost of its children nodes. For OR nodes it is an average of the costs for the children nodes, weighted by the probability assigned to each child.

The function $Eval(GR, PRT_\alpha, C_\alpha)$ represents the expected utility to the group for carrying out the recipes represented in PRT_α in context C_α. It is the difference between the expected utility to the group for carrying out α and the expected cost, given the recipes represented in PRT_α. It combines the expected utility of the parent node with the expected utilities of its children. The expected utility of a node is the value of the action that the node represents multiplied by the success probability ($p\text{-}CBA$) of the node. If the node is an OR node, the expected utility of each child node is weighted by its branching probability. Computation of the *Eval* function requires traversing the entire PRT, because a recipe that is selected for one of the subactions in a PRT may affect the evaluation of a recipe for another subaction in the tree.

The $Select\text{-}PRT(G_1, \alpha, C_\alpha)$ refers to the PRT that represents G_1's belief about the possible recipes it will select to perform action α in context C_α. $Predict\text{-}PRT(G_1, G_2, \alpha, C_\alpha)$ refers to the PRT that represents G_1's belief about the possible recipes G_2 will select to perform action α in context C_α. The PRT given for the appetizer in Figure 1, represents Alice's belief for the recipes that Bob may select for making the appetizer. In addition, the predicate $Selected\text{-}PRT(G_1, PRT_\alpha, C_\alpha)$ is true if PRT_α is the recipe that is selected by G_1 in context C_α.

Type	Notation	Meaning
Actions	$\alpha \in \Omega$	top level action
	$\beta \in \Omega$	$\beta \in$ *sub-actions*(α)
	$\gamma \in \Omega$	helpful act
Agents	$GR \subset \mathcal{A}$	agents involved in SharedPlan for α
	$G_1 \in GR$	agent reasoning about helpful behavior
	$G_2 \in GR$	partner(s) of G_1
Time	T_i	current time
	T_α	time of execution for α
Contexts	$C_\alpha \in \mathcal{C}$	$Context(C_\alpha, G_1, \alpha, T_i)$
	$C_\alpha^{T_\alpha} \in \mathcal{C}$	$Context(C_\alpha^{T_\alpha}, G_1, \alpha, T_\alpha)$
	$C_{GR} \in \mathcal{C}$	$Bel(G_1, Context(C_{GR}, GR, \alpha, T_i))$
	$C_{GR}^{T_\alpha} \in \mathcal{C}$	$Bel(G_1, Context(C_{GR}^{T_\alpha}, GR, \alpha, T_\alpha))$
	$C_\beta \in \mathcal{C}$	$Context(C_\beta, G_1, \beta, T_i)$
	$C_\beta^{Bel} \in \mathcal{C}$	$Bel(G_1, Context(C_\beta^{Bel}, G_2, \beta, T_i))$
	$C_\gamma \in \mathcal{C}$	$Context(C_\gamma, G_1, \gamma, T_i)$
$\mathcal{PRT}$	$PRT_\alpha \in \mathcal{PRT}$	PRT selected for action α
	$PRT_\beta \in \mathcal{PRT}$	PRT selected for action β
	$PRT_\gamma \in \mathcal{PRT}$	PRT selected for action γ

Table 2: Summary of notations

5. HELPFUL BEHAVIOR MODELS

In this section we present a decision-theoretic mechanism containing rules that agents can use to decide whether to undertake helpful behavior. We consider two types of helpful behavior: communicating with a partner and performing a specific domain action to assist a partner. Both types refine the recipes agents use, but in different ways. An agent that decides to communicate information to a partner will cause the partner to update its context to reflect this information, which may lead this partner to adopt a new recipe that is more likely to succeed. An agent that performs a helpful domain action is doing so to increase the likelihood that the partner's recipe, and by extension the group activity, will succeed. Table 2 presents the notation that is used in specifying the decision-theoretic mechanism.

5.1 Commitment to Helpful Behavior

In this section, we define an agent's commitment to an activity in a way that reflects the agent's intention that a recipe that is optimal for the group (given the agent's beliefs) is selected. The clause $Committed(G_1, GR, \alpha)$ refers to the commitment of agent G_1 to the success of the group GR for achieving α, when there exists a recipe PRT_α that G_1 believes will maximize the group utility, and G_1 intends that all group members intend to carry out PRT_α at execution time. In Definition 1 – the definition of $Committed$ – $SP(GR, \alpha, C_\alpha)$ is true if GR has a SharedPlan for α in C_α, Bel is the standard modal operator for representing beliefs, $Selected\text{-}PRT(GR, PRT_\alpha, C_{GR}^{T_\alpha})$ is true if it is believed that PRT_α is selected by GR at time T_α in $C_{GR}^{T_\alpha}$, $\mathcal{PRT}_{-\alpha}$ is the set of PRTs for α excluding PRT_α.

Definition 1 : Committed(G_1, GR, α), is true if agent $G_1 \in GR$ is committed to GR's success in doing α

Committed(G_1, GR, α) iff
$[(G_1 \in GR) \wedge \text{SP}(GR, \alpha, C_\alpha) \wedge$
$(\exists PRT_\alpha \in \mathcal{PRT})$
$[\text{Bel}(G_1, (\forall PRT_i \in \mathcal{PRT}_{-\alpha})$
$[\text{Eval}(GR, PRT_i, C_\alpha^{T_\alpha}) \leqslant \text{Eval}(GR, PRT_\alpha, C_\alpha^{T_\alpha})]) \wedge$
$\text{Int.Th}(G_1, \text{Selected-PRT}(GR, PRT_\alpha, C_{GR}^{T_\alpha}))]]$

5.2 Performing a Helpful Action

The process for making the decision whether to perform a helpful action γ is specified by Algorithm 2. The first conditional establishes that agent G_1 is committed to the success of the group for achieving α. The second conditional holds if G_1 believes that it can improve the utility of the action by executing the helpful act γ. To compute the utility of adding helpful action γ, G_1 reasons about the recipes it will adopt for helpful act γ, and the improvement it can generate by adding these recipes to the ones selected by GR for action α. Agent G_1 intends to do γ, if it believes that doing γ will lead to an improvement in the group's utility for carrying out α.

Algorithm 2 : Helpful-Act(G_1, GR, α, γ), G_1 helps GR in doing α by doing γ if doing so increases the expected utility of GR for doing action α, and G_1 is committed to GR's success in doing α

```
if Committed(G1, GR, α) then
    PRT_α := Predict-PRT(G1, GR, α, C_GR)
    PRT_γ := Select-PRT(G1, γ, C_γ)
    PRT_α^Help := PRT_α ⋃ PRT_γ
    utility := Eval(GR, PRT_α^Help, C_α) - Eval(GR, PRT_α, C_α)
end if
if utility > 0 then
    Int.To(G1, γ, C_γ)
end if
```

For example, if Bob believes Alice may intend to make pasta with tomato sauce, but he knows there are no fresh tomatoes left in the kitchen, he can perform a helpful action by adopting an intention to go to the market and buy some tomatoes. He should do so only if the cost related with going to the market is lower than the potential benefit to the dinner they are making.

5.3 Deciding to Communicate

The ability to communicate information allows agents to convey changes in the world or to request information about the world. We present two rules for deciding whether to communicate.

5.3.1 Conveying Information

In situations in which two agents G_1 and G_2 are committed to the success of a collaborative activity, when G_1 makes an observation, it needs to reason about informing G_2 about this observation. The decision to communicate may improve the utility of the group, but communication is associated with a cost. Algorithm 3 specifies the process by which G_1 reasons about this trade-off. In particular, G_1 reasons about the recipes that G_2 would adopt for doing sub-action β if G_1 has communicated observation o. If the utility gain to the group from the adoption of this recipe is higher than the cost of communication, then G_1 will communicate o to G_2.

In algorithm 3, $Comm(G_1, G_2, o)$ refers to the inform action, $COC(G_2)$ represents the cost of communicating with G_2. $Context\text{-}Update(C_\beta^{Bel}, o)$ represents context C_β^{Bel} updated with observation o.

Algorithm 3 : Inform(G_1, G_2, α, β, o), G_1 informs G_2 about observation o, if doing so increases the expected utility of GR for α, and G_1 is committed to GR's success in doing α, where G_2 has intention to do β

```
if Committed(G1, GR, α) then
    PRT_α := Predict-PRT(G1, GR, α, C_GR)
    C_β^o := Context-Update(C_β^Bel, o)
    PRT_β^o := Predict-PRT(G1, G2, β, C_β^o)
    PRT_α^o := PRT_α ⊗ PRT_β^o
    utility := Eval(GR, α, PRT_α^o, C_α) −
               Eval(GR, α, PRT_α, C_α)
end if
if utility > COC(G2) then
    Int.To(G1, Comm(G1, G2, o), C_α)
end if
```

For example, if Bob sees that there are no tomatoes in the kitchen, and he thinks that Alice is making a tomato sauce, he would conclude that their meal-making is likely to fail. If he informs Alice about this observation, Alice can update her recipe so that it does not contain tomatoes. If Bob forecasts that the utility improvement generated by Alice updating her recipe given the observation is higher than communication costs, Bob will inform Alice. However, if Bob believes that Alice is likely not to be using tomatoes or the communication cost is very high, then he would not inform Alice.

5.3.2 Asking for Information

Agent G_1 may need to reason about asking G_2 for ψ, if this information would beneficially change G_1's recipe for doing sub-action β. To compute the utility of asking G_2 about ψ, G_1 needs to consider how it will adapt its own belief about the recipes it will select, for each possible answer that is provided by G_2. Agent G_1 computes the difference in expected group utility for α between its initial belief about recipes to select for α and any refined belief that is a result of the answer given by G_2.

In algorithm 4, C^o_α and C^o_β are contexts updated by G_1 with answer o, $Comm(G_1, G_2, \psi)$ refers to the ask action, $COC(G_2)$ represents the cost of communicating with G_2, $\Psi(\psi)$ is the set of observations that are answers to ψ, $pr(Bel(G_2, o))$ is G_1's prediction of the probability of receiving o from G_2 as an answer to ψ. G_1 believes there exists an observation $o \in \Psi(\psi)$ that G_2 believes to be the correct answer to question ψ.

Algorithm 4 : Ask(G_1, G_2, α, β, ψ), G_1 committed to doing β, asks G_2 question ψ, if doing so increases the expected utility of GR's doing α, and G_2 is committed to GR's success in doing α

if Committed(G_2, GR, α) **then**
  PRT_α := Predict-PRT(G_1, GR, α, C_{GR})
  for $o \in \Psi(\psi)$ **do**
    C^o_β := Context-Update(C_β, o)
    C^o_α := Context-Update(C_α, o)
    PRT^o_β := Select-PRT(G_1, β, C^o_β)
    PRT^o_α := $PRT_\alpha \otimes PRT^o_\beta$
    utility := utility + $pr(Bel(G_2, o)) \times$
      (Eval(GR,α,PRT^o_α,C^o_α) −
        Eval(GR, α, PRT_α, C^o_α))
  end for
end if
if utility > $COC(G_2)$ **then**
  Int.To(G_2, Comm(G_1, G_2, Ψ), C_α)
end if

For example, if Alice has a plan for making tomato sauce but believes that her plan may fail as a result of there being no good tomatoes, she can ask Bob if he knows the availability of tomatoes. For each possible answer Alice may receive from Bob, she updates her belief about recipes to select that incorporates that answer. After weighting each possible updated recipe with the probability of receiving that answer, Alice computes the expected utility for asking. If it is higher than the communication cost, Alice considers asking. However, if Alice believes that the answer will not improve the recipe she selects, or the cost of communication is very high, then she would not consider communicating with Bob.

The extent to which ask decisions are profitable depends on how well agents are able to model how the world changes. If G_1 believes the world is uncertain but it is not, it will keep asking needlessly. If G_1 is not expecting any changes but the world is changing, it fails to ask when needed.

The complexity of Algorithm 4 depends heavily on the size of $\Psi(\psi)$, the number of observations that are answers to question ψ. Recent work has provided techniques to facilitate this computation for situations with large numbers of possible answers in collaborative settings [12].

6. EMPIRICAL EVALUATION

In this section we provide an empirical evaluation of the mechanism described in Section 5. The evaluation used the Colored Trails (CT) system, a publicly available test-bed developed for investigating decision-making in task settings, where the key interactions are among goals, tasks required to accomplish those goals, and resources needed to perform the tasks [1] [7]. CT is played on a rectangular board of colored squares. The players are located randomly on the board. They are given chips of the same colors used in the game board. Goal squares are positioned in various locations on the board, and the object of the game is to reach those goals. At each turn of the game agents can move to an adjacent square on the board by surrendering a chip in the color of the square.

Our experiments used a configuration of CT in which certain squares on the board may turn into traps and prevent players' advancement. Players have full visibility of the board and players' positions on the board, but cannot observe the chips the other player has. One of the players (called the observer) is able to observe the trap locations, whereas the other player (the partner) cannot. The game proceeds for a specified number of turns. At the end of the game, a score is computed for each player that depends on the number of goals the player was able to achieve, the number of chips left in its possession, and the score for the other player.

This CT game is an analogue of a task-setting in which players have partial knowledge about the world and about each other. Chips represent agents' capabilities, and a path to a goal square on the board represent agents' plans for achieving their goals. There may be several paths agents can take to reach a goal square, like there are several recipes that agents can use to achieve their goal. Traps represent the possibility that a plan may fail in a world that may change. Players have an incentive to collaborate in this game because their scores depend on each other.

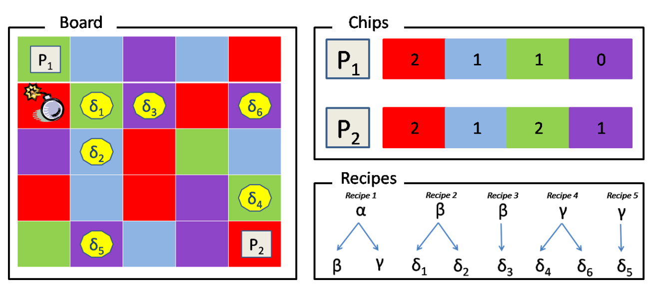

Figure 2: Screen-shot of CT game

A snapshot of the game is presented in Figure 2. The game board is displayed on the left of the figure; P_2 represents the observer player and P_1 represents its partner; δ_1 to δ_6 represent lower-level goals that are positioned on the board itself; α, β and γ represent higher level goals. Players' chips are shown on the top-right of the figure, and the possible recipes for α are presented on the bottom-right section of the figure.

In this game, P_1 is committed to accomplishing goal β because it is closer to the constituents goals of β, and P_2 is committed to γ. P_1 is able to achieve goal β by achieving sub-goals δ_1 and δ_2, but it cannot achieve δ_3 because it lacks a purple chip. P_1 is unable to observe trap positions, and a trap is located just below its current location.

In this CT game, agents' context include both their beliefs about the world (e.g., probability distribution over possible trap positions), and beliefs about their partner's contexts (e.g., probability distribution over the chips the partner possesses). The success probability (p-CBA value) of a path

[1] CT is open source software and can be downloaded at http://www.eecs.harvard.edu/ai/ct

towards a goal is obtained by combining these beliefs. A player successfully moves to an adjacent square if it is not a trap position. The value of the basic action *cba.basic* corresponds to the probability that the square the agent moves onto is not a trap. Players receive 100 points for reaching the goal (completing task α), and 10 points for reaching any of δ_i. Thus the valuation is 100 points for α, 10 for any δ_i, and zero for the rest of the goals. The agreed upon utility function of our formalism corresponds to the joint scoring function of the game. A sample PRT for a collaborative plan for the CT game is shown in Figure 3.

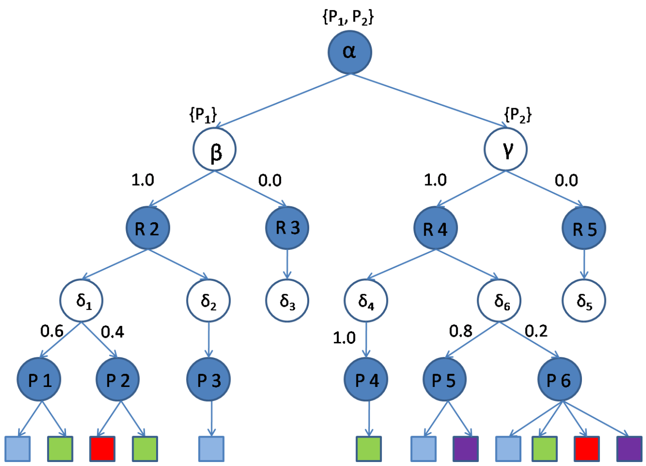

Figure 3: A sample PRT for CT SharedPlan game

In the game, the observer can help its partner by giving away chips so that the partner is able to realize a path to the goal which was formerly inaccessible. In addition, players can communicate information about traps in one of two ways: the observer can inform its partner about trap positions, or its partner can ask about the location of traps directly. There is a cost associated with all of these helpful-behavior actions, and players need to weigh this tradeoff when they engage in helpful-behavior decisions.

6.1 Experimental Setup and Results

We used the CT game described above as a test-bed for quantitative analysis of our decision-theoretic helpful behavior models. The helpful behavior rules are evaluated in terms of the average score they generated across 500 game plays. The significant differences in average scores of the protocols are tested with t-test for paired two-samples for means, and labeled whenever the difference was not in the 95% confidence interval. In all experiments, the chips agents possess in the game and the board layout are drawn from a uniform distribution that is common knowledge between players. The probability that a trap may appear for a given color is known to the observer, but not to its partner.

The first set of experiments compared the following three protocols for deciding whether to perform helpful act: The Helpful Act protocol uses Algorithm 2 to determine whether to perform a helpful act; the Random Help protocol gives away a random colored chip; the No Help protocol never gives away chips. We varied whether the observer has complete or incomplete information about the chips of its partners. Figure 4 shows the joint scores that players achieve by using the different protocols. The results show that the

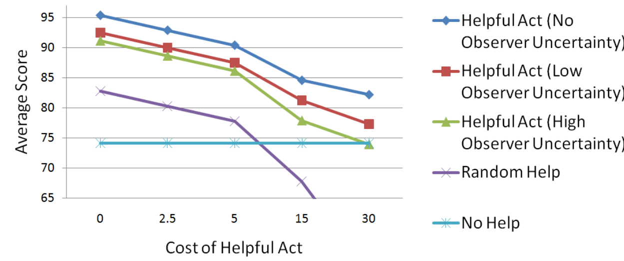

Figure 4: Performance of helpful act protocols by helpful act cost, observer uncertainty

Helpful Act protocol performs significantly better than the Random Help and the No Help protocols. Except for cost 30, the performance of Helpful Act with high observer uncertainty is not significantly different than No Help. The performance of the Helpful Act protocol improves significantly as the observer's uncertainty decreases.

The second set of experiments compared four different communication protocols. The Inform protocol uses Algorithm 3 to determine whether the observer should tell its partner about trap positions. The Ask protocol utilizes Algorithm 4 to determine whether the partner should ask the observer about trap positions. In the Always Inform protocol (AI), the observer always informs its partner, regardless of the its partner's need. The Never Communicate protocol (NC) does not allow any type of communication. We varied three factors to assess the performance of these protocols: the communication cost, observer uncertainty about the partner's chips, and the uncertainty in the world about how frequently traps occur on the board. As the partner player's uncertainty about the world increases, we expected the benefit of communication to increase because the partner cannot observe trap positions.

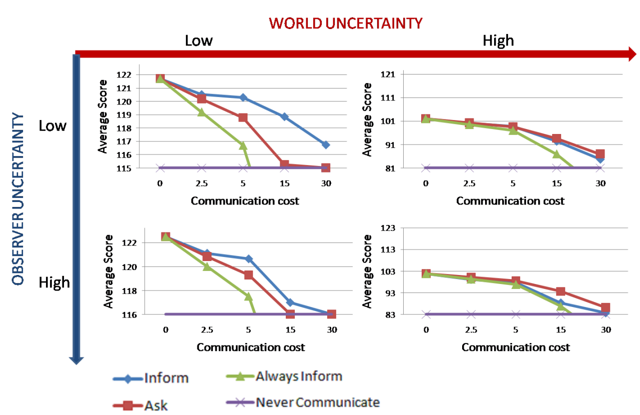

Figure 5: Performance of communication protocols by communication cost, observer uncertainty, world uncertainty

Figure 5 shows the average performance of the different communication protocols. On each graph, the vertical axis represents the average score of the game; the horizontal axis

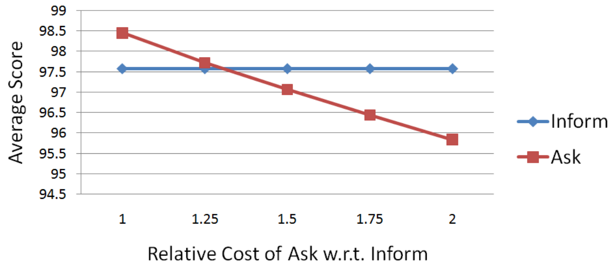

Figure 6: Performance of Inform and Ask protocols as relative cost of Ask w.r.t Inform varies (high observer and world uncertainty, communication cost is 5).

varies the cost of communication from low to high. The four graphs cover four possible configurations of world and observer uncertainties (low-low, low-high, high-low, high-high). As shown in the figure, the decision-theoretic protocols (Ask and Inform) outperform or perform as well as the AI and NC protocols for all communication costs and uncertainty levels. When the observer has a good model of its partner (observer uncertainty is low), the Inform protocol performs better than (or equally as good as) the other communication protocols because the observer gets to see the (sometimes unexpected) changes in the world and is good at predicting when its observations are useful for its partner. Interestingly, when world uncertainty is high, the partner expects the world to change frequently and benefits from asking the observer about traps; therefore the Ask protocol performs better or equivalent to other protocols. However, when the traps happen to change position and the world uncertainty is low, the Inform protocol is better. Overall, the decision-theoretic protocols outperform axiomatic (i.e., non-decision theoretic models without probabilistic representation) models. The performance of the two decision-theoretic models varies with the uncertainty conditions

So far we have assumed that the relative communication costs of Ask and Inform protocols are identical. However, the cost of the Ask protocol may be higher than the Inform protocol because the Ask protocol includes two steps of communication; from the partner to the observer and from the observer to the partner. Figure 6 shows the average performance of Ask and Inform protocols given that the relative cost of Ask with respect to Inform varies from 1.0 (identical) to 2.0 (double the communication cost).

7. CONCLUSION AND FUTURE WORK

We have presented a new decision-theoretic mechanism for managing helpful behavior in collaborative settings. Our mechanism considers the uncertainty associated with the domain, beliefs about the recipes selected by other group members, and the cost entailed by helpful behavior to reason about the utility of helpful behavior and to determine whether help should be given. We introduced a compact and efficient representation for modeling agents' belief about recipes selected by other agents. This representation is used to efficiently update and evaluate joint plans. Our work focused on two distinct types of helpful behavior: communicating and adding a helpful act to the group plan. We conducted empirical evaluation of our helpful behavior mechanism in different collaborative settings that varied the level of uncertainty and the cost of helpful behavior. Our results show that our decision-theoretic mechanism performed better than axiomatic methods in all of these cases.

In ongoing work, we are exploring: enriching the helpful behavior mechanism with rules about who, when and how to help; considering a chain of communication between team of agents; extending the helpful behavior mechanism to the collaboration of self-interested agents. This work has relevance to research in multi-agent systems, planning, team formation and domain modeling for all settings on which agents can help.

8. ACKNOWLEDGMENTS

The work reported in this paper was supported in part by NSF grant IIS-0705406 and in part by DARPA Contract No. FA8750-05-C-0033.

9. REFERENCES

[1] P. R. Cohen, H. J. Levesque, and I. Smith. On team formation. In *Contemporary Action Theory. Synthese*, pages 87–114. Kluwer Academic Publishers, 1997.

[2] F. Dignum and H. Weigand. Modelling communication between cooperative systems. In *CAiSe '95: Proceedings of the 7th International Conference on Advanced Information Systems Engineering*, pages 140–153, London, UK, 1995. Springer-Verlag.

[3] X. Fan, J. Yen, and R. Volz. A theoretical framework on proactive information exchange in agent teamwork. *Artificial Intelligence*, 169(1):23–97, 2005.

[4] C. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[5] B. Grosz and L. Hunsberger. The dynamics of intention in collaborative activity. *Cognitive Systems Research*, 7(2-3):259–272, 2006.

[6] B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[7] B. Grosz, S. Kraus, S. Talman, B. Stossel, and M. Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. *AAMAS*, 2004.

[8] N. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75(2):195–240, 1995.

[9] D. Kinny, M. Ljungberg, A. Rao, E. Sonenberg, G. Tidhar, and E. Werner. Planned team activity. *Artificial Social Systems*, 1992.

[10] H. Levesque, P. Cohen, and J. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 94–99. Boston, MA, 1990.

[11] M. Pollack. Plans as complex mental attitudes. *Intentions in Communication*, pages 77–103, 1990.

[12] D. Sarne and B. J. Grosz. Estimating information value in collaborative multi-agent planning systems. In *AAMAS*, page 48, 2007.

[13] M. Tambe. Towards Flexible Teamwork. *Journal of Artificial IntelligenceResearch*, 7:83–124, 1997.