Between collaboration and competition: An Initial Formalization using Distributed POMDPs

Praveen Paruchuri
Computer Science Department
University of Southern California
Los Angeles, CA 90089
paruchur@usc.edu

Milind Tambe
Computer Science Department
University of Southern California
Los Angeles, CA 90089
tambe@usc.edu

Spiros Kapetanakis
Department of Computer Science
University of York, UK
kapetana@usc.edu

Sarit Kraus
Department of Computer Science
Bar-Ilan University
Ramat-Gan 52900, Israel
Institute for Advanced Computer Studies
University of Maryland
College Park 20742
sarit@macs.biu.ac.il

ABSTRACT

This paper presents an initial formalization of teamwork in multi-agent domains. Although analyses of teamwork already exist in the literature of multi-agent systems, almost no work has dealt with the problem of teams that comprise self-interested agents.

The main contribution of this work is that it concentrates specifically on such teams of self interested agents. Teams of this kind are common in multi-agent systems as they model the implicit competition between team members that often arises within a team. Our work models the internal struggle of agents that are acting in a team as they try to maximise their individual payoff while at the same time acting in a manner that is beneficial to the entire team. This dilemma of self interest versus team interest is a problem that has been studied in game and decision making theory, although no clear-cut solution that applies to agent systems has been proposed.

Our formalisation is based on the theory of Partially Observable Markov Decision Processes (POMDPs). In this work, we reintroduce and extend the Electric Elves (E-Elves), an application of personal assistant agents that displays all the characteristics of competition within a cooperative setting. Using E-Elves we show how competition arises out of a collaborative scenario and analyse the shortcomings of previous approaches in handling this competition. Finally, we provide some initial thoughts on how to cope with these problems based on our previous experience with E-Elves.

1. INTRODUCTION

A large number of recent applications have focused on enabling software agents (and/or robots) to assist humans in their organizational activities, such as in offices, educational institutions, research organizations, business organizations[4, 26, 29, 9]. The motivation in these applications is that agents embedded in human organizations can enable routine organizational coordination, flexible response to a changing environment or rapid response to crises. Initial prototypes of such applications have already been developed and in some instances deployed, and this area has begun to see a rapid growth.

In these applications, agents must often act on behalf of individual human users, groups of human users or even on behalf of critical resources in the organization. These agents must thus ensure that their user's goals are appropriately achieved, and their interests are appropriately protected. Yet, agents within such human organization must act together jointly in organizational activities, or else the organizational objectives may be difficult to attain. For instance, even in simple tasks such as scheduling or rescheduling meetings for their users (e.g., see [26]), agents must ensure that their user's preferences are honored and yet, these agents coordinate with each other and reach joint agreements.

The focus of this paper is to take some initial steps towards a formal computational model of the type of domain mentioned above. The main characteristic of agents in these domains is that while they belong to a team or to the same organization and may have some joint goals, they also have their own personal goals and must maximize their own benefits in the context of the team or organization. There are many models of individual agents acting in competitive environments and attempting to maximize their personal benefits [20]. There are also several models of teamwork of agents that act together and attempt to maximize the overall benefits of the team, founded on BDI logics[5, 8], and decision theory[21]. However, almost no work has focused on modelling teams of self-interested agents, although there are some notable exceptions[27]. In such teams, an agent needs to balance between its desire that the teamwork will succeed while, at the same time, protecting its own (or its user's) interests.

This paper presents EMTDP (extended multiagent team decision problem) as a formal computational model for such teams of self-interested agents. EMTDP is based on previous models of distributed POMDPs[15], in particular COM-MTDP[21]. However, there are some major novelties in EMTDP. First, unlike the distributed POMDPs that focus on team activities by modelling a single joint reward for the team, EMTPD maintains the joint reward for the team and the individual rewards for the individual participants in the team. Thus, rewards occurring from joint activities are both the joint rewards and the individual rewards.

A further critical novelty in EMTDP is in its solution concept. In distributed POMDPs, an optimal policy is one which attempts to maximize the joint value. This optimal policy is appropriate in pure teamwork situations where there is only a single joint reward. However, such an optimal policy could potentially lead to arbitrarily low individual rewards for a team member, when constructing teams of self-interested agents. Such low rewards may provide the individual an incentive to deviate from the stated policy. Instead, in EMTDP, the optimal policy must also ensure certain minimum expected value for individual team members.

To provide a practical illustration of the benefits of the EMTPD model, we focus on a real-world application called *Electric-Elves*[26] (E-Elves). In previously published results[26], this application used a fully collaborative model; indeed a single MDP with a single joint reward was used to model the actions of agents in E-Elves. We illustrate that this model leads to difficulties precisely because the notion of self-interest within a team was not (and could not be) appropriately modelled. We illustrate that using EMTPD enables us to resolve the prior difficulties encountered in the E-Elves system.

2. EMTDP

This section describes the Extended Multiagent Team Decision Problem (EMTDP) model. It also provides an analysis of the model's ability to represent the important aspects of multi-agent teamwork in domains such as the E-Elves, where cooperation and competition exist simultaneously. The EMTDP model is an extension of the Multiagent Team Decision Problem (MTDP)[22]. We first describe MTDP briefly and then introduce the extension to EMTDP.

2.1 MTDP

An MTDP is a tuple $< S, A_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R >$, where we assume a team α of n agents $\alpha_1, \alpha_2, \ldots, \alpha_n$. We explain each term below:

- S is a set of world states, expressed as a cross product of separate features. In other words, S is the state of the team's environment.
- A_α is the set of allowed actions for all agents in the system. $\{A_i\}_{i\in\alpha}$ is a set of domain-level actions for each agent $\imath$ to perform in the environment which is known as the agent's *action space*. These actions implicitly define the set of combined (or team) actions $A_\alpha \equiv \prod_{i\in\alpha} A_i$
- $P : S \times A_\alpha \times S \to [0,1]$ is a probability distribution that governs the effects of domain-level actions. For each initial state s at time t, combined action a taken at time t and final state s' at time $t+1$ we have $Pr(S^{t+1} = s'|S^t = s, A^t_\alpha = \alpha) = P(s, a, s')$
- $\{\Omega\}_{i\in\alpha}$ is a set of observations that each agent α_i can experience of the world. The combined observation is defined implicitly as $\Omega_\alpha \equiv \prod_{i\in\alpha} \Omega_i$.
- O_α is a joint observation function. Typically O_α is defined as the cross product of each agent's observation function: $O_\alpha \equiv \prod_{i\in\alpha} O_i$ where $O_i(s, \alpha, \omega) = Pr(\Omega^t_i = \omega|S^t = s, A^{t-1}_\alpha = \alpha)$
- Each agent $\alpha_i \in \alpha$ forms a belief state $\beta^t_i \in B_i$ based on its observations seen through time t, where B_i circumscribes the set of possible belief states for the agent. Implicitly, the set of possible combined belief states is defined as $B_\alpha \equiv \prod_{i\in\alpha} B_i$
- A *common* reward function R for the team is defined as $R : S \times A_\alpha \to R$. The reward function represents the team's joint preferences over the states and the cost of domain-level actions.

A policy for an agent in MTDP is any function $\pi : B_i \to A_i$ that maps the agent's belief state to an action or probability distribution over many actions in its action space. The objective of MTDP is to find joint policies $(\pi_1, \pi_2, \ldots.., \pi_n)$ such that these joint policies provide a maximum expected reward.

2.2 Extension to EMTDP

As noted in section 2.1, the reward function in the MTDP model is centralized. This is due to the fact that, in MTDP, all agents are assumed to have the same preferences. However, this is not always the case and, in domains such as the E-Elves, it is often necessary to allow agents to have different preferences.

To describe the agents' preferences and accurately reward the agents for their actions, the reward function is transformed into a sum of terms, each of which represents a specific reward to each of the agents in the team. Thus there is a joint team reward, but also components of the joint reward which are individual rewards available to each team member of the team as a result of the joint action, denoted as $< R_{priv_1}, ..., R_{priv_n} >$

The flow of rewards to the agents is as follows: at each point in time, all agents decide what actions to perform based on their beliefs at the time. The *joint* action that is made up from all the agents' individual actions is executed. The system then transitions to the corresponding successor state and all agents receive a joint reward equally. In addition, the agents each receive a private reward, where the private terms are calculated for each agent individually and the corresponding rewards are distributed where due. So, for every transition from one state to another via a joint action, all agents get the same joint reward from the reward function. It is only in their private components that the agents' rewards differ.

Thus, more than one private terms often coexist in the extended version of the reward function. However, for simplicity we assume that the joint reward is a weighted sum of the private rewards. Thus, we will henceforth use the function $R = W1 * R_{\text{priv1}} + W2 * R_{\text{priv2}}$, where R_{priv1} and R_{priv2} are the private terms and $W1$ and $W2$ are weights.

3. THE SOLUTION CONCEPT

In section 2.1, we described the notion of a policy in the MTDP framework. The same notion is carried across to the extended model since, again, a policy is simply a function that prescribes what action an agent should (potentially probabilistically) undertake based on its belief state. However, our goal in the EMTDP model is to find one policy π_i per agent such that the expected joint reward for the team is maximized under the constraint that, for all policies π_i, the expected private reward for π_i is higher than the threshold V_{min}. Thus agents attempt to optimize team performance. The manner in which these policies are found is through the use of a policy generator. We iteratively generate *all* possible policies for the agents. All combinations of these policies must be evaluated in order to find the set of policies that satisfy the algorithm's criteria. We now potentially have a number of sets of policies for all agents which satisfy the algorithm's constraint. One of those is selected at random as the solution to the problem.

The reason for the V_{min} constraint on the private reward is that the agents in EMTDP have different preferences. This means that, if after a policy search, each agent is given a policy to follow that is optimal for the team, it is likely that some of the agents will get a significantly lower private reward than others due to differences arising from their private reward terms i.e. differences due to their preferences over the possible outcomes.

The fact that the agents have different preferences may mean that, in some cases, there simply isn't a solution that is optimal for all the agents simultaneously. With the added constraint, optimality is relaxed so that the agents can reach a consensus after policy generation with the guarantee of minimum private reward. The philosophy behind the constraint is that it is beneficial for the agents to agree to a course of action that guarantees them a minimum private reward even if other agents accumulate higher private rewards. The bias in the agents' decision making is towards agreeing since the agents are in a team, although they also have interests of their own. The need for this constraint never materialized in the original MTDP framework since there was only one central reward function whose optimal point was optimal for all agents simultaneously.

Imagine, for example, the situation where there are two possible sets of policies that are accepted by all agents, namely the set $(\pi_1, \pi_2, \ldots, \pi_n)$ and the set $(\rho_1, \rho_2, \ldots, \rho_n)$. More specifically, let us assume that agent i prefers outcome π and agent j prefers outcome ρ. By preference we mean that, although both sets of policies are acceptable since their i_{th} and j_{th} component's satisfy agents i and j respectively (i.e. provide expected reward higher than V_{min}), individually policy π_i yields higher private reward for agent i whereas policy ρ_j yields higher private reward for agent j. The problem of selecting one of the two sets of policies is resolved by the agents' agreement that a policy is acceptable as long as it yields reward higher than the threshold V_{min}. So, random choice between the two policy sets is satisfactory for both agents.

It is critical to note that in the domains of interest, agents do not benefit from unilaterally deviating from the policy provided, if the policy is based on the constraints mentioned above. In particular, in these domains, the actions are fundamentally joint actions towards joint goals, and thus, an agent cannot perform such actions alone. For instance, the next section discusses the example of a meeting; here, an agent cannot meet along by itself, and thus deviating from the provided policy — as long its interests are protected to a certain minimum level — is not to the agent's benefit.

4. EXPERIMENTAL RESULTS

This section presents our experimental results and a hypothesis aimed at validating the claims made in the previous section. The E-Elves system introduced earlier was used as a testbed for evaluating the E-MTDP model. Rather than constructing a new toy example, the goal here was to use a previously published, real-world system as a testbed for EMTDP. We first describe the E-Elves domain more extensively and then proceed to the experimental results. Our claim is that the absence of models such as the E-MTDP cause difficulties in creating E-Elves like systems and thus such problems can be mitigated via E-MTDP.

4.1 Modelling Elves MDP as an EMTDP

The Electric Elves (E-Elves) was a project at USC/ISI to deploy an agent organization in support of the daily activities of a human organization [23, 3]. Teams of software agents can aid organizations in accomplishing these tasks, facilitating coherent functioning and rapid, flexible response to crises. In the E-Elves system, each user's personal assistant agent acts on behalf of the user in the agent team. These agents are called *proxies*. The proxy can perform several tasks for its user. For example, if a user is delayed to a meeting, the proxy can reschedule the meeting, informing other proxies, who in turn inform their users. If there is a research presentation slot open, the proxy may respond to the invitation to present on behalf of its user. The proxy can also order its user's meals and track the user's location, posting it on a Web page. The proxies communicate with their users using wireless devices, such as personal digital assistants (PALM VIIs) and WAP-enabled mobile phones, and via user workstations.

We focus here on the task of rescheduling meetings. The original E-Elves system used a single agent MDP to model the actions of an individual agent in the team. The actions available to the agent include delaying the meeting, cancelling the meeting, asking the user for input, suggesting to the team that the meeting go ahead without the user and so forth. Asking for user input is critical in E-Elves, as the proxy agents are taking actions on behalf of the users and must consult with them; yet time constraints from the rest of the team may prohibit an agent from asking for user input so that sometimes autonomous actions are taken instead. The rest of the agent team typically follows the recommendations made by this single-agent MDP say in delaying the meeting (although there is a simple filtering process to determine if the recommendations should be followed or not by the team — we will ignore these cases for now).

Thus, in many instances, the single agent's MDP could be viewed as a team MDP, modelling joint actions, joint rewards and joint states. The original joint reward function of the MDP is discussed below [26]. We later illustrate how this single reward function can be subdivided to model an EMTDP.

$$
\begin{aligned}
R(s,a) \;=\; & \sum_{entity \in E\setminus\{A\}} EQ^{decision}_{entity}(time(s)) \cdot \\
& \textit{entity-response} \\
& -\lambda_1 f_1(MeetingDelay) \\
& -\lambda_2 f_2(late,h) \\
& +\lambda_3 r_\alpha(\textit{value of meeting without user}) \\
& +\lambda_4 r_{user}(\textit{user's individual value to } \alpha) \\
& +\lambda_5 f_3(a)
\end{aligned} \qquad (1)
$$

We proceed to explain the terms present in the above joint reward function and provide their classification into *private1* and *private2* terms. Here *private1* refers to one agent's individual reward. While *private2* refers to the other individual's reward, where the second individual is the other team member, i.e., the other meeting attendee in the case of E-Elves. Since the other meeting attendee may actually be more than one individual, we refer to this term as *private2*. (However, as before, the joint reward seen above is a weighted sum of the *private1* and *private2* rewards). The first component captures the value of getting a response from a decision making entity other than the agent itself, i.e. this is the reward that a proxy agent obtains when it asks a user for input and actually obtains a response from the user. Only one entity responds to the request and the reward is associated with the agent that receives that response. Hence this term is a *private1* reward term.

The f_1 function reflects the inherent value of attending the meeting as the team originally expected. This deters the agent from making any costly coordination changes (meeting delays) unless they can gain some indirect value from doing so. This term is a *private2* reward since it is a penalty to the other team member (the other team member was able to meet on time, but now must delay the meeting).

The f_2 function corresponds to the cost of making the meeting attendees wait. By definition it includes the cost of other team members and can be classified as *private2* .

The component r_α models the inherent value of the joint activity, i.e., the meeting. It represents the value of the meeting,if it takes place, but the user does not attend. This component can be classified as a *private2* term because it invloves the other team members (excluding the user). The value r_{user} models the user's individual value to α. This represents the value added to the meeting by the user and is classified as *private1* part of the reward. The f_3 function accounts for additional costs of tranfer of control actions. Since communication is between the agent and the user it accounts for *private1* part of the reward.

4.2 Preliminary experiments

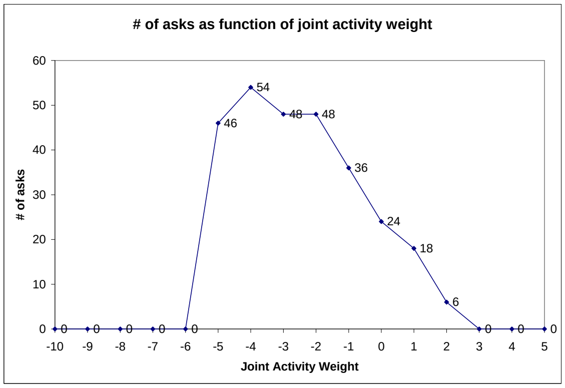

Figure 1: Electric elves evaluation

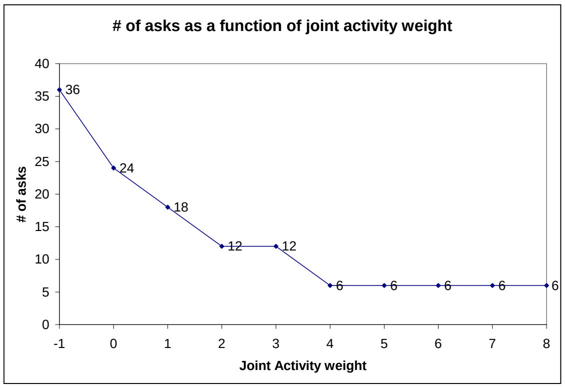

Figure 2: Electric elves evaluation

As we pointed out earlier, one of the challenges of the E-Elves system is taking the decision of whether to ask the user or not. In this experiment the value of the meeting without the user (i.e r_α) is varied and the number of asks has been obtained, as shown in Figure 1.

We have chosen to plot the number of asks against the value of the meeting We can see from the plot that, as the joint activity weight varies from -10 to -6 the number of asks start increasing. The plot reaches a peak at -4 and again starts decreasing till it reaches zero at 3.

The reason for the shape of the curve is that when the value of the meeting is set to too low value, the agent tries to minimize communication costs associated with giving a quality decision about the user's view. Therefore, it tries to make autonomous decisions. On the other hand when the value of the meeting is set at a high value the agent cannot afford the uncertainty, in the user providing response in time and hence tries to take decisions autonomously. Only in the intermediate regions the agent tries to ask the user. In the first instance, the agent doesn't want to bother the user with a decision about a relatively unimportant meeting whereas, in the second instance, the agent cannot afford to ask the user because the cost of potentially not getting a response in time is too high.

Though this looks reasonable logically, the behaviour of the agent would look very counter-intuitive to the user. In particular when the meeting becomes important the user is not even asked his opinion before the agent replies to the team. This was one of the biggest problems that was faced by the E-Elves system when it was put to practical use. In most of the situations the user would like to be asked a certain number of times before any important decision is made.

The problem arises because the original Elves focused on only maximizing the team's joint reward, and did not focus on the self-interest of each team member, e.g., the user. Thus, this problem is mitigated to a great extent due to the EMTDP model. One of the main features of the EMTDP model is that it ensures certain minimum expected reward for individual team members. The main reason why the agent takes decisions autonomously is that it gives undue importance to the rest of the team members (the "private2" terms unnecessarily carry much higher weight). By making the individual important to the agent, the agent is now forced to ask users opinion also.

The reward function in the EMTDP framework is of the form $reward = \gamma * private1 + \delta * private2$, where, γ and δ are constants between 0 and 1. The policy generator satisfies the constraint that expected reward is maximized while it always maintains a value above a certain threshhold. This ensures that the number of asks cannot fall below a certain minimum number making the system more predictable and dependable. The important issue would be setting the threshhold which depends on the user's choice. Figure 2 shows the effect of maintaining such a threshhold. (Please note that we have not fully implemented the algorithm mentioned in Section 3; instead we have simulated its results by changing internal parameters).

From the above arguments we conclude that the EMTDP model is novel because it has provision for maintaining two conflicting rewards. It further ensures that the user has the flexibility to tune the agent according to his tastes as seen in the new E-Elves system.

5. RELATED WORK

Some of the problems that arise in environments such as the E-Elves can be modelled, using game theory techniques, as *coordination games*. A coordination game is a game in which there is at least one outcome which both agents prefer over other outcomes. The problem arise when there are several outcomes that are preferred by all parties over others; however, the agents have conflicting preferences among these preferred outcomes. For example, consider a situation of two users of the E-Elves application who need to set up a meeting. They need to choose a meeting time on Tuesday or Wednesday. One user prefers meeting on Tuesday and the second prefers meeting on Wednesday. However, both prefer to meet in any of these days rather than to cancel the meeting. Thus, both choosing Tuesday is an equilibrium but also both choosing Wednesday is an equilibrium. The problem is which of these equilibria will be chosen.

Equilibrium selection theories have been developed for such games which can discriminate between Nash equilibria (e.g., [10, 12, 6]). A significant amount of work has been performed on the evolution of equilibria in coordination games that are played repeatedly within a population (e.g. [32, 31, 13, 2, 7, 14].) However, there is no acceptable solution for one shot games or for complex situations as occur in the E-Elves domain.

The tension between an individual and a group has been studied in the economics and game-theory literature also in the context of an agent that subcontract a task to a group of self-interested agents (e.g. [17, 18, 16]). The problem of the contracting agent is to provide the contracted agents with an incentive to make a costly effort that will contribute to the success of the task even when the contracting agent can observe only the final outcome of the agents' activity but can't observe each of the agents' personal effort (e.g. [11, 24, 1]). In such situations, the well known free rider problem arises: each of the agents would like the others to work hard, and would like to minimize its own effort. The solution is based on the design of complex contracts that are offered by the contracting agent. That is, there is a central manager that plays an important role in this setting. We consider situations where such central agent is not available.

As mentioned above, vast array of work has been performed on competitive multi-agent systems or on cooperative multi-agent systems. In some models, it is assumed that the competitive agents will cooperate but usually no formal motivation is given for their cooperation.

We will demonstrate the problems of such approaches in looking closely, as an example, at the technically interesting paper of Vickrey and Koller [30]. They consider the problem of collaboratively finding a stable strategy profile in situations involving multiple interacting agents. Their main example is the following. Suppose a road is being built from north to south through undeveloped land and $2n$ agents have purchased plots of land along the road the agents $w_1, ..., w_n$ on the west side and the agents $e_1, ..., e_n$ on the east side. Each agent needs to choose what to build on his land; a factory, a shopping mall, or a residential complex. His utility depends on what he builds and on what is built north, south, and across the road from his land. All of the decisions are made simultaneously. The agents in this example are not cooperative; each tries to maximize its own expected utility.

The special property of the above game is that it is an example of a graphical game. It assumes that each agent's reward function depends on the actions of a subset of the agents rather than on all other agents' actions. They use this property when searching for an equilibrium. In particular, they focus on the idea of exploiting the locality of interaction between agents, using graphical games as an explicit representation of this structure. They provide two algorithms that exploit this structure to support solution algorithms that are both computationally efficient and utilize distributed collaborative computation that respects the "lines of communication" between the agents. However, it is not clear why the agents will collaborate. Their agents are competitive. It is clear that once an equilibrium is identified and all the agents agree to follow their identified equilibrium strategies, there is no incentive to each of them to deviate. However, since each of the agents may prefer a different equilibrium, it is not clear why they will not deviate during the search for the strategy, instead of following the proposed search algorithm.

Grosz et al. [28] considered the problem of teams of self-interested agents. They focused on intention reconciliation in a team context. That is, the agents need to reconcile their intentions to do team-related actions with other, conflicting, but more beneficial, intentions. This problem was studied empirically using the SPIRE simulation. They assumed that a centralized agent makes the initial task allocation, while we look for a theory that will enable a team of self-interested agents to agree upon task allocation and to make any other decisions that are needed for their joint activity.

Gmytrasiewicz[25] proposes a novel formulation for distributed POMDPs, where instead of a centralized planner, there are separate agents planning their own optimal policies for their own POMDPs individually. While this is a very interesting advance, the concept of optimal policies is not fully elaborated. In particular, precisely how would or should different agents pick their optimal policies without negotiations with other agents is unclear. Finally, Nair et al[19] use the concept of a Nash equilibrium to speed up the search for an optimal policy in MTDP. However, they focus on a single joint reward rather than the individual rewards that are also considered in EMTDP.

6. DISCUSSION

In the original electric elves domain the policy generation was done in a distributed fashion. There was a centralized creation of joint rewards based on general rules. By dividing the reward into private1(which is t he personal reward) and private2(the team reward) in the E-MTDP the agents had the flexibility to set their threshholds. However, in our implementation we held the threshholds fixed for all the agents. We would like our model to handle the case where each individual can specify its own threshholds. There is however a serious problem with the individuals setting their threshholds. It can happen that in such a free to set threshhold scenario the agents can start setting their threshholds high. This leads to the free rider problem where agents try to preserve their pe rsonal interests at the cost of team.

One issue that was raised by readers is that there is no need for a separate individual and joint reward, since a joint reward with appropriately adjusted weights could be used to solve an equivalent problem.In other words, EMTDP could be subsumed by MTDP.However we tend to argue a bit differently. Let us consider a simple example as follows:

$4,4$	$0,0$	$0,0$
$0,0$	$100,1$	$0,0$
$0,0$	$0,0$	$1,100$

In the table above, the tuple (value 1, value 2) say (4,4) refers to the private reward of agent 1 and private reward of agent 2. Therefore in the perspective of agent 1, value 1 equals private1 and value 2 equals private2 and for agent 2, value 1 equals private2 and value 2 equals private1.

In this example if the threshhold was 5, there is no possible solution. However, maximizing the reward function would have to choose one of them as them as a solution. This shows that our method is inherently different from just adjusting the weights of different terms and maximize the reward function. Suppose the threshhold is 3. In this case selecting (4,4) is the best move according to the E-MTDP policy. There is no way in which weights can be adjusted so as to select (4,4) in the reward maximaization method.

The other advantage of such a division is the added understandability, modifiability and explainability. Instead of dealing with each and every individual term in the reward function such a split is a very clean approach of fine tuning behaviour of the system. There is a clearly defined part of the reward that the user can adjust without bothering about details of the actual terms in the reward function.

7. CONCLUSIONS

This paper presents an initial formalization of multiagent teamwork, where multiple agents each may have some self interest which they must protect, while simultaneously aiming to attain the team goal. Such teamwork is critical in newly emerging domains where agents are embedded in human organizations, and must protect the interests of the human users while also acting in a team.

Our initial attempt has yielded the formulation of EMTDP, where agents have both a team reward and individual rewards; EMTDP is an extension to a distributed POMDP framework called MTDP. We introduce a criteria for selecting policies in EMTDP.

Finally, we experiment with Electric Elves (E-Elves), an existing application with published results. We illustrate that by using a single joint team reward, this application may suffer from unpredictability in agents' behaviors. If instead, this application was modeled as an EMTDP, some of these difficulties could be addressed. These initial results illustrate the potential benefits of EMTDP for realistic future applications.

8. REFERENCES

[1] A. Banerjee and A. Beggs. Efficiency in hierarchies: implementing the first-best solution by sequential actions. *The Rand Journal of Economics*, 20(4):637–645, 1989.

[2] V. Bhaska. Breaking the symmetry: Optimal conventions in repeated symmetric games. In *The 17th Arne Ryde Symposium on Focal Points: Coordination, Complexity and Communication in Strategic Contexts*, Sweden, 1997.

[3] H. Chalupsky, Y. Gil, C. Knoblock, K. Lerman, J. Oh, D. Pynadath, T. Russ, and M. Tambe. Electric Elves: Applying agent technology to support human organizations. In *Proceedings of International Conference on Innovative Applications of AI (IAAI-01)*, pages 51–58, 2001.

[4] Hans Chalupsky, Yolanda Gil, Craig A. Knoblock, Kristina Lerman, Jean Oh, David V. Pynadath, Thomas A. Russ, and Milind Tambe. Electric Elves: Agent technology for supporting human organizations. *AI Magazine*, 23(2):11–24, 2002.

[5] Philip R. Cohen and Hector J. Levesque. Teamwork. *Nous*, 25(4):487–512, 1991.

[6] R. W. Cooper, D. V. DeJong, R. Forsythe, and T. W. Ross. Selection criteria in coordination games: Some experimental results. *The American Economic Review*, 80(1):218–233, 1990.

[7] V. P. Crawford and H. Haller. Learning how to cooperate: Optimal play in repeated coordination games. *Econometrica*, 58:571–595, 1990.

[8] Barbara Grosz and Sarit Kraus. Collaborative plans for complex group actions. *Artificial Intelligence*, 86:269–358, 1996.

[9] Kautz H.A., B. Selman, Coen M., Ketchpel S., and Ramming C. An experiment in the design of software agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, August 1994.

[10] J. C. Harsanyi and R. Selten. *General theory of equilibrium selection in games*. MIT Press, Cambridge, Mass, 1988.

[11] B. Holmstrom. Moral hazard in teams. *Bell Journal of Economics*, 13(2):324—340, 1982.

[12] John B. Van Huyck, Raymond, C. Battalio, and Richard O. Beil. Tacit coordination games, strategic uncertainty, and coordination failure. *The American Economic Review*, 80(1):234–248, 1990.

[13] M. Kandori, G. J. Mailath, and R. Rob. Learning, mutation, and long run equilibria in games. *Econometrica*, 61(1):29–56, 1993.

[14] F. Kramarz. Dynamic focal points in N-person coordination games. *Theory and Decision*, 40(3):277–313, 1996.

[15] M.L. Littman L.P. Kaelbling and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101, 1998.

[16] C. Ma. Unique implementation of incentive contracts with many agents. *Review of Economic Studies*, 51(3):555–572, 1988.

[17] C. Ma, J. Moore, and S. Turnbull. Stopping agents from "cheating". *Journal of Economic Theory*, 46:355–372, 1988.

[18] D. Mookherjee. Optimal incentive schemes with many agents. *Review of Economic Studies*, 51(3):433—446, 1984.

[19] Ranjit Nair, Milind Tambe, and Stacy Marsella. Role allocation and reallocation in multiagent teams: Towards a practical analysis. *AAMAS*, 2003.

[20] M. J. Osborne and A. Rubinstein. Games with procedurally rational players. *American Economic Review*, 88:834–847, 1998.

[21] David V. Pynadath and Milind Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.

[22] David V. Pynadath and Milind Tambe. Multiagent teamwork: Analyzing the optimality and complexity of key theories and models. *AAMAS*, 2002.

[23] David V. Pynadath, Milind Tambe, Hans Chalupsky, Yigal Arens, et al. Electric elves: Immersing an agent organization in a human organization. In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*, 2000.

[24] E. Rasmusen. Moral hazard in risk-averse teams. *Rand Journal of Economics*, 18(3):324—340, 1987.

[25] Bharaneedharan Rathnasabapathy and Piotr J. Gmytrasiewicz. Multi-agent pomdp's in the context of network routing. *Proceedings of the 36th Hawaii International Conference on System Sciences*, January 2003.

[26] Paul Scerri, David V. Pynadath, and Milind Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228, 2002.

[27] David G. Sullivan, Barbara J. Grosz, Sarit Kraus, and Sanmay Das. The influence of social norms and social consciousness on intention reconciliation. *Artificial Intelligence journal*, 142(2):147–177, 2002.

[28] David G. Sullivan, Barbara J. Grosz, Sarit Kraus, and Sanmay Das. The influence of social norms and social consciousness on intention reconciliation. *Artificial Intelligence journal*, 142(2):147–177, 2002.

[29] K. Sycara and D. Zeng. Coordination of multiple intelligent software agents. *International Journal of Cooperative Information Systems*, 5(2,3), 1996.

[30] D. Vickrey and D. Koller. Multi-agent algorithms for solving graphical games. In *Proc. of AAAI-02*, 2002.

[31] P. Young. The evolution model of bargaining. *Journal of Economic Theory*, 59:145–168, 1993.

[32] P. Young. The evolution of conventions. *Econometrica*, 61(1):57–84, 1993.

[33] M. Rosemary Emery. Game-Theoretic Communication Policies for Partially Observable Markov Games. *Thesis Proposal*, 2003