An MDP model for planning team actions with communication

Yuqing Tang
Department of Computer Science
Graduate Center, City University of New York
365, 5th Avenue, New York, NY 10016

Simon Parsons
Department of Computer & Information Science
Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210

November 19, 2009

1 Introduction

In our work in Project 10, Task 1, we are investigating how agents can help teams to better perform their allocated tasks. Our work to date has developed models that construct plans for the members of a team, plans that allow the team to reach some specified team goal, and then extract from this the individual actions that team members have to perform. A key element of our work has been factoring into the plans, and the model that constructs the plans, the need for communication. In particular, we recognise that because the members of a team only have a very partial view of the team's progress, team members will often be uncertain as to what their best next step is and that appropriate and timely communication to resolve this kind of ambiguity has the potential to greatly improve team performance.

The models that we have developed to date (for example [6]) have been symbolic models, adapted from the literature of non-deterministic planning. Here we take an additional step, moving from the world of non-deterministic planning to the world of decision-theoretic planning, and in particular developing a model for planning the actions and the communications of a team using Markov Decision Processes (MDPs).

2 MDP model

Following the literature on decision theoretic planning [3, 4], we can define an MDP as a tuple

$$\mathcal{M} = \langle \mathcal{P}, S, A, Pr, R \rangle$$

where

- $\mathcal{P} = \mathcal{P}_S \times \mathcal{P}_A$ is a finite set of propositions;
- $S = 2^{\mathcal{P}_S}$ is the set of all possible states;
- $A = 2^{\mathcal{P}_A}$ is the finite set of actions;
- $\Pr$ is a probability distribution on S conditional on $S \times A$ so that $Pr_a(s, s')$, where $a \in A$ and $s, s' \in S$, is the probability of an agent arriving in state s' when it takes action a in state s; and
- R is a reward distribution on S conditional on $S \times A$ so that $R_a(s, s')$ where $a \in A$ and $s, s' \in S$ is the immediate reward the agent gets when it is in state s and takes the action a and ends up in state s'. The reward can be further decomposed into

  $$R_a(s, s') = U_a(s, s') - C_a(s, s')$$

  where

  - $U_a(s, s')$ is the immediate utility an agent gets when it is in state s and takes the action a, resulting in being in state s'; and
  - $C_a(s, s')$ is the cost to the agent of taking action a in s to get to s'.

We diverge a little from the standard MDP model here in including a propositional language as part of the model. We need this because we are going to use MDPs to describe both the actions of an agent in the world, and the action of an agent in terms of communication, and the propositional language will allow us to relate these two models together.

3 Policies

3.1 What is a policy?

We solve MDP models to get a *policy*, a description of what to do in each state of the model. In particular, we consider a policy to be a set of state-action pairs,

$$\pi = \{\langle s_i, a_i \rangle\}$$

where $s_i \in S$ and $a_i \in A(s)$ with

$$A(s) = \{a | \exists s' P_a(s, s') \neq 0\}$$

The usual notion of a policy, as in [3] includes one action for each state, so the choice of action in that state is easy. Because of our focus on planning for teams, we think of policies being constructed in terms of what we term *joint actions*, in other words actions specify what is done by every member of the team[1], and which can then be decomposed to give actions for each individual agent.

A fundamental difference between our work and other work in this area is that although we assume that in each state there is at most one joint action in the policy, it may not be the case that each agent has a unique action for each state of the team as whole. In other words an agent that, for example is taking part in a flanking operation, might not be able to distinguish between several team states for which it has different actions — states in which it should attack immediatel and states in which it should wait before attacking for example. This occurs because in many situations an agent only has local knowledge of its own state, and without information from other agents cannot resolve the ambiguity about which action to perform. In effect, then, the policy for the team — what we will call the *joint policy* — specifies more than one action for a given agent for a given state of that agent. We call a policy that does this *non-deterministic*.

In general, a policy π is a non-deterministic iff there is a state s such that π prescribes more than one action. When non-determinism present, and no way to resolve it, agents choose each action in π with equal probability. By overloading the notation, $\pi(s)$ is used to represent the set of actions associated with s in π

$$\pi(s) = \{a | \langle s, a \rangle \in \pi\}.$$

Moreover, $\pi(s, a)$ is used to represent the action selection probability:

$$\pi(s, a) = \frac{1}{|\pi(s)|}.$$

In the case when π is deterministic in the state s, $\pi(s, a) = 1$. Below we will consider at length how to use communication to handle non-deterministic policies.

A *history* is a sequence of state-action pairs that ends with a state. For example, $h = s_1, a_1, \ldots, s_{n-1}, a_{n-1}, s_n$ is a history. A history $h = s_1, a_1, \ldots, s_{n-1}, a_{n-1}, s_n$ can be induced from a policy π, denoted by $h \in induced(\pi)$, if every pair $\langle s_i, a_i \rangle \in h$ is also in π and $Pr_{a_i}(s_i, s_{i+1}) > 0$. Given the Markov property,

$$Pr_\pi(h = s_1, a_1, \ldots, s_n) = \prod_{i=1}^{n-1} Pr_{a_i}(s_i, a_i) \cdot \pi(s_i, a_i).$$

The reward of a history is defined as follows

$$R(h = s_1, a_1, \ldots, s_n) = \sum_{i=1}^{n-1} \gamma^{i-1} R_{a_i}(s_i, a_i)$$

where $\gamma \in [0, 1]$ is the discount factor (which guarantees convergence to a finite reward even though the policy execution may be infinite).

[1] These actions do not necessarily involve team members performing parts of a larger action — like helping to lift a heavy object — merely they involve team members all having a specified thing to do, which may, of course, be “do nothing”.

The goal of policy planning is then looking for a policy π that can maximize the expected long term utility for the system for all possible histories $s_1, a_1, \ldots, s_\infty$ induced from π in:

$$argmax_\pi \sum_{h \in induced(\pi)} Pr_\pi(h) \cdot R(h)$$

Two special cases of policy planning are

- Given a start state s_0, produce a policy $\pi(s_0)$ such that the execution structure under π has the maximum expected utility

  $$R_\pi(s_0) = Max_\pi \left[\sum_{h \in induced(\pi), h=[s_0 \ldots]} (Pr_\pi(h) \cdot R(h)) \right]$$

- Given a set of start states S_0, produce a policy $\pi(S_0)$ such that the execution structure under π has the maximum expected utility over all the start states $s \in S_0$

  $$R_\pi(S_0) = Max_\pi \sum_{s \in S_0} Pr(s) Pr_\pi(h) R(h)$$

3.2 The execution structure

One view of a policy is that it is the MDP equivalent of a plan — a choice of actions to be executed, though a plan that, because there is an action for every possible state, covers every possible eventuality. Another way to think of a policy is in terms of the states that it connects through the nominated actions. This leads us to the concepts of *transition graph* and *execution structure*.

Definition 1 *The transition graph of an* MDP $\mathcal{M} = \langle \mathcal{P}, S, A, Pr, R \rangle$ *under a policy* π *is a graph* $Tr = \langle S, E \rangle$ *where*

- S *are the nodes of the graph, each of which corresponds to a state in the* MDP.
- E *are the edges in the graph,each of which corresponds to a state-state transition enabled by the policy* π.

  Each edge $\langle s, s' \rangle \in E$ *is labeled by a set of triples* $\{\langle a, pr, r \rangle\}$ *with each element* $\langle a, pr_a, r_a \rangle$ *specified as follows*

  - $a \in A$ *is an action that satisfies* $Pr_a(s, s') \neq 0$ *and* $\langle s, a \rangle \in \pi$,
  - $pr_a = Pr_a(s, s')$ *is the corresponding probability, and*
  - $r_a = R_a(s, s')$ *is the corresponding reward.*

Combining the concepts of MDPs and the execution structure of non-deterministic state transitions we have:

Definition 2 *The execution structure of an* MDP $\mathcal{M} = \langle \mathcal{P}, S, A, Pr, R \rangle$ *under a policy* π *is a graph* $\Sigma = \langle S, E \rangle$ *where*

- *S are the nodes in the graph, each of which corresponds to a state in the* MDP*.,*
- *$E \subseteq S \times S$ are the edges in the graph, each of which correspond to a state transitions enabled by the policy π.*

  Each edge $\langle s, s' \rangle \in E$ is labeled by a set of triples $\{\langle h, pr_h, r_h \rangle\}$ defined as following

  - *$h = s_1, a_1, \dots, s_n$ is the history which be induced from π with $Pr(h) > 0$ and $R(h)$, $s_1 = s$, and $s_n = s'$.*
  - *$pr_h = Pr(h)$*
  - *$r_h = R(h)$*

4 The Joint MDP model

4.1 Joint actions

To describe the behavior of a team, we need to prescribe more structure over the actions available to an agent. We assume that in the system there is a set of N agents labeled by $\mathcal{T} = \{T_1, T_2, \dots, T_N\}$ each of which is modeled by MDP

$$\mathcal{M}_i = \langle \mathcal{P}_i, S_i, A_i, Pr_i, R_i \rangle.$$

We call the actions in the set A the *joint actions* of these agents.

In the MDP literature which has addressed multiagent systems, for example, [1], the name "Decentralized Markov Decision Process" is often used for this situation. We prefer the name "joint MDP" partly because it conforms with our previous work on non-deterministic dialogue policy planning and partly because it emphasizes our focus on building plans for teams — we are more interested in the fact that the MDP describes joint actions than the fact that it is decentralized (though of course it is decentralized, and DEC-MDPs deal with joint action).

The joint MDP model is then

$$\mathcal{M} = \langle \mathcal{P}, S, A, Pr, R \rangle$$

where

- $\mathcal{P} = \cup_i \mathcal{P}_i$,
- $S = \Pi_i S_i$,
- $A = \Pi_i A_i$, and
- Pr and R satisfy the properties specified below.

In the joint model, each action $a \in A$ is a tuple of actions of individual agents, so $a = [a_1, \dots, a_n]$. That is each action $a \in A$ can be further decomposed into n actions $a_i \in A_i$ of individual agents T_i. Based on the underlying propositional logic, we also write $a \models a_i$ if agent T_i's action is a_i in a joint action a, $s \models s_i$ if agent T_i's perception of a (joint) state s is s_i.

We can introduce additional conditions β on agents, and corresponding probability and reward constraints Pr_β and R_β, can be introduced. Together these specify the logical, probabilistic, and reward inter-dependencies between the state and action variables of different agents. A form of logical interdependency can be found in as in our previous work [6], and we will study the forms of the probability and reward inter-dependencies in our future work.

With these notions, now we can specify the conditions on the relation between the joint probabilities and rewards and those of the individual agents:

- $Pr_{a_i}(s_i, s'_i) = \sum_{\{\langle s,a,s'\rangle | s \models s_i, s' \models s'_i, a \models a_i\}} Pr_a(s, s')$:

  The local transition probabilities are obtained by marginalizing over all the states and actions that can be mapped to the same local states.

- $R_{a_i}(s_i, s'_i)$

  Local transition rewards will conform with some application dependent criteria and assumptions. One example is the assumption that the joint utility is the sum of individual utilities and a system wide reward in which case:

  $$R_a(s, s') = \Sigma_{i=1\ldots N} R_{a_i}(s_i, s'_i) + R_{a|sys}(s, s')$$

  where $s = [s_1, \ldots, s_N]$, $s' = [s'_1, \ldots, s'_N]$, $a' = [a'_1, \ldots, a'_N]$, $R_{a_i}(s_i, s'_i)$ is agent T_i's reward, and $R_{a|sys}(s, s')$ is a system wide reward.

In addition, we will have the following notions to map joint states, actions, and policies to those in their agent T_i's local model

- $proj_i(s) = \{s_i | s \models s_i\}$ is the projection of a joint state to a set of local states of agent T_i;

- $proj_i(a) = \{a_i | a \models a_i\}$ is the projection of a joint action to a set of local actions of agent T_i;

- $proj_i(\pi) = \{\langle s_i, a_i\rangle | \langle s, a\rangle \in \pi, s \models s_i, a \models a_i\}$ is the projection of joint policy to a local policy (most likely to be a non-deterministic one) of agent T_i.

4.2 The shadow joint policy

We do not anticipate that individual agents will hold a full copy of the set of joint states and joint actions. Even if an agent knows the joint MDP model and the joint policy, as discussed above its limited view of the world means that it cannot typically carry out the policy exactly. Instead what the team of agents do is to carry out an approximation of the joint policy in which each agent makes the best choices it can given the information available to it. We call the resulting policy that is executed the *shadow joint policy*. This is a useful concept in establishing the efficiency with which a team of agents manages to carry out a policy.

Informally, the *shadow joint policy* of a joint policy π is an implicit joint policy in which each agent makes random choices from the set of actions it can project for its current state from the joint policy π. Before we can define the shadow policy precisely, we need a concept of similarity which captures the fact that agents cannot tell certain joint states apart.

Definition 3 *Two joint states s and ss are similar in agent T_i's view, denoted by $s \sim_i ss$, iff there exists an $s_i \in S_i$ such that*

- $proj_i(s) = s_i$*, and*
- $proj_i(ss) = s_i$*.*

Similarly, two joint actions a and aa are similar in agent T_i's view, denoted by $a \sim_i aa$, iff there exists an $a_i \in A_i$ such that

- $proj_i(a) = a_i$*, and*
- $proj_i(aa) = a_i$*.*

Thus two states are similar for a given agent if they map to the same local state, and two joint actions are similar if they map to the same action for that agent.

Definition 4 *A joint policy π's shadow policy is a joint policy $\pi^{\approx}$ whose entries are of the form $\langle ss, aa \rangle$ with $aa = [a_1, a_2, \ldots, a_N]$ where for each agent T_i ($i = 1, \ldots, N$) there exists an entry $\langle s, a \rangle \in \pi$ such that*

- $s \sim_i ss$*, and*
- $a_i = proj_i(a)$

The corresponding shadow execution *of a joint policy π can be defined as*

$$\Sigma^{\approx}(\pi) = \Sigma(\pi^{\approx})$$

5 The MDP communication model

At this point we have a formal model that is sufficiently rich to construct plans that involve the physical actions that agents carry out. However, we want to create plans that include communications that permit the necessary sharing of information, so we need to add a communication model to the model we already have. We refer to this model as a *dialogue* model in recognition of the fact that our long-term goal is to extend the model so that it permits agents to engage in complex communications.

As the basis of the dialogue model, we will use the same kind of MDP model as we use for the world model. To distinguish the two state transition models, we will denote these two models and their elements with subscripts. We write $_{|D}$ to denote elements of the dialogue model, for example, $M_{|D}$ denotes the state transition model for a dialogue and $S_{|D}$ denotes the states of a dialogue. We write $_{|W}$ to denote elements of the world model, for example, $M_{|W}$ denotes the external world model and $S_{|W}$ the states of the world. However, when the state transition model is obvious from the context, we will omit the subscripts.

As before, we assume that, in the dialogue, there is a set of N agents labeled $T_1, T_2, \ldots, T_N$ where each agent T_i has a model of the world:

$$\mathcal{M}_{i|W} = \langle \mathcal{P}_{i|W}, S_{i|W}, A_{i|W}, Pr_{i|W}, R_{i|W} \rangle$$

and for which it has a policy $\pi_{i|W} = \{\langle s_i, a_i \rangle\}$. Given this, a dialogue model is then a state transition system:

$$\mathcal{M}_{|D} = \langle \mathcal{P}_{|D}, S_{|D}, A_{|D}, Pr_{|D}, R_{|D} \rangle$$

for which there is a policy for conducting dialogues $\pi_{|D}$. Correspondingly, each agent T_i has an individual dialogue model:

$$\mathcal{M}_{i|D} = \langle \mathcal{P}_{i|D}, S_{i|D}, A_{i|D}, Pr_{i|D}, R_{i|D} \rangle$$

and dialogue policy $\pi_{i|D}$.

The dialogue language $\mathcal{P}_{|D}$ contains elements from language $\mathcal{P}_{i|W}$ that individual agents use to describe the world, along with auxiliary language elements such as a proposition to mark the differences between two world states. The dialogue information is induced from $\mathcal{P}_D$. The set of dialogue acts $\mathcal{A}_{|D}$ are those available to the agents. We will give a concrete example in the following sections.

As before, a policy for a dialogue, $\pi_{|D} = \{\langle s_{|D}, a_{|D} \rangle\}$, specifies what dialogue actions should be taken in a given dialogue state. To distinguish such policies from the policies that govern an agent's actions in the world, we call the policies that govern an agent's actions in a dialogue a *conversation policy* and a policy that governs an agent's actions in the world a *world policy*.

Definition 5 *Agent T_i's behavior model with dialogue transitions added is a tuple*

$$\mathcal{M}_i = \langle \mathcal{P}_i, S_i, A_i, Pr_i, R_i, F_{i|W \to D}, F_{i|D \to W} \rangle$$

where

- $\mathcal{P}_i = \mathcal{P}_{i|W} \cup \mathcal{P}_{i|D}$*,*
- S_i *and* A_i *are induced from* $\mathcal{P}_i$ *as before,*
- Pr_i *is a probability distribution,*
- R_i *is a reward distribution,*
- $F_{i|W \to D}$ *maps the external world states and actions to dialogue states, and*
- $F_{i|W \to D}$ *maps the dialogues to world states and actions*

The probability distribution Pr_i captures both external world transitions and dialogue transitions as follows:

- External world transitions:

  $$Pr_{\langle a_{i|W}\rangle}(\langle s_{i|W}, s_{i|D}\rangle, \langle s'_{i|W}, s'_{i|D}\rangle) = Pr_{a_{i|W}}(s_{i|W}, s'_{i|W})$$

  where

  - $a_{i|W} \in F_{i|D\to W}(s_{i|D})$ is the world action selected after communication.
  - $s'_{i|D} = F_{i|W\to D}(s'_{i|W})$ picks the corect dialogue state after a world action.

- Dialogue transitions:

  $$Pr_{\langle a_{i|D}\rangle}(\langle s_{i|W}, s_{i|D}\rangle, \langle s_{i|W}, s'_{i|D}\rangle) = Pr_{a_{i|D}}(s_{i|D}, s'_{i|D})$$

- Otherwise, Pr_i is set to 0.

Note that all the above are conditional probabilities on behavior state-action pairs. This along with the Markov property will make Pr_i a valid probability distribution.

The reward function R_i also has to handle external world transitions and dialogue transitions. It does this as follows:

- External world transitions:

  $$R_{\langle a_{i|W}\rangle}(\langle s_{i|W}, s_{i|D}\rangle, \langle s'_{i|W}, s'_{i|D}\rangle) = R_{a_{i|W}}(s_{i|W}, s'_{i|W})$$

  where

  - $a_{i|W} \in F_{i|D\to W}(s_{i|D})$ is the world action selected after communication
  - $s'_{i|D} = F_{i|W\to D}(s'_{i|W})$: resets the dialogue state after the world action

- Dialogue transitions:

  $$R_{\langle a_{i|D}\rangle}(\langle s_{i|W}, s_{i|D}\rangle, \langle s_{i|W}, s'_{i|D}\rangle) = R_{a_{i|D}}(s_{i|D}, s'_{i|D})$$

- Otherwise, R_i is set to 0.

The functions $F_{i|W\to D}$ and $F_{i|D\to W}$ define the relationship between world states and actions and the dialogue states of individual agents. $F_{i|W\to D}$ is actually two functions

- First there is a function to map a world state to a set of dialogue states:

  $$F_{i|W\to D} : S_{i|W} \to 2^{S_{i|D}}$$

  This also comes in a version that operates on a set of states:

  $$F_{i|W\to D}(S) = \cup_{s\in S} F_{i|W\to D}(s)$$

- Second there is function to map a world action to a set of dialogue states

  $$F_{i|W\to D} : A_{i|W} \to 2^{A_{i|D}}$$

  This also comes in a version that operates on a set of states:

  $$F_{i|W\to D}(A) = \cup_{a\in A} F_{i|W\to D}(a)$$

Sometimes, we will overload the notion to map a world state-action pair to a set of dialogue states

$$F_{i|W \to D} : S_{i|W} \times A_{i|W} \to 2^{S_{i|D}}$$

where $F_{i|W \to D}(s_{i|W}, a_{i|W}) = F_{i|W \to D}(s_{i|W}) \cap F_{i|W \to D}(a_{i|W})$.

$F_{i|D \to W}$ is also composed of two functions

- First there is a function that maps a dialogue state to a set of joint world states:

  $$F_{i|D \to W} : S_{i|D} \to 2^{S_{|W}}$$

  This also comes in a version that operates on a set of states:

  $$F_{i|D \to D}(S) = \cup_{s \in S} F_{i|D \to W}(s)$$

- Second, there is a function that maps a dialogue state to a set of joint world action

  $$F_{i|D \to W} : S_{i|D} \to 2^{A_{|W}}$$

  This also comes in a version that operates on a set of states:

  $$F_{i|D \to W}(S) = \cup_{s \in S} F_{i|D \to W}(s)$$

In addition we may overload the notion to map a dialogue states to a joint world state-action pairs:

$$F_{i|D \to W} : S_{i|D} \to 2^{S_{|W} \times A_{|W}}$$

where $F_{i|D \to W}(s_{i|D}) = F_{i|D \to W}(s_{i|D}) \times F_{i|D \to W}(a_{i|D})$

What we have so far specifies the world and dialogue models of each agent. We can then put these together to get an MDP that describes the whole multiagent system. We have:

Definition 6

$$\mathcal{M} = \langle \mathcal{P}, S, A, Pr, R \rangle$$

where

- $\mathcal{P} = \cup_{i=1}^{N} \mathcal{P}_i$
- *S and A are induced from $\mathcal{P}_i$ as before*
- *Pr is composed from the Pr_i of individual agents.*

  To do this we can assume independence among agents or specify more complex interactions using logical constraints β, probability constraints Pr_β and reward constraints R_β.
- *R_i is composed by adding individual agents' rewards together*

6 A detailed look at the communication model

The reason for including communication in the model is so that we can identify when agents can and should pass information to their teammates in order to help those teammates to disambiguate states and make better choices of action. As a result, the dialogue state transition is about world states, actions and their executions. The ontology of $\mathcal{P}_{|D}$ will be built on top of $\mathcal{P}_{|W}$ and some additional control variables to ensure that the dialogue state transitions can enable the construction of a dialogue policy which can prescribe effective dialogue actions to a set of dialogue states that result from applying $F_{i|W \to D}$ on the external states of each individual agent T_i.

6.1 An ontology for dialogues

Recall that, in general, $\mathcal{P}_{i|D} = \mathcal{P}_{i,S|D} \cup \mathcal{P}_{i,A|D}$.

We start by assuming that each agent T_i maintains a model of the external world $\mathcal{M}_{i|W}$ and its finite propositional language $\mathcal{P}_{i|W}$ will depend on the application. T_i's dialogue model $\mathcal{M}_{i|D}$ is based on a propositional language:

$$\mathcal{P}_{i,S|D} = \cup_{j=1}^{N}(\mathcal{P}_{j,S|W} \cup \mathcal{P}_{j,A|W}) \cup \mathcal{P}_{AL} \cup \mathcal{P}_{CM} \cup \mathcal{P}_{ACT}$$

where:

- $\mathcal{P}_{AL}$ contains a boolean variable for every variable in $\mathcal{P}_{j,S|W} \cup \mathcal{P}_{j,A|W}$ for each $j = 1, \ldots, N$ to indicate its validity in dialogue state.
- $\mathcal{P}_{CM}$ contains a boolean variable for every variable in $\mathcal{P}_{j,S|W} \cup \mathcal{P}_{j,A|W}$ of the agent T_i's state (the information T_i can send to ther agents) to indicate whether its value has been communicated in dialogue state j, $j = 1 \ldots N$ where N is the number of agents in the system.
- $\mathcal{P}_{ACT} = \{z_j | j = 1 \ldots N\}$ with $z_j = 1$ indicates the agent T_j should act in the external world, and $z_j = 0$ indicates that the agent T_j should continue the dialogue.

In the simple model of communication that we consider here agents can only inform each other of information that the sender believes to be true. As a result, dialogue actions will be of the form:

$$tell(i, j, x_k, v)$$

and the locations will be encoded by the following set of proposition variables

$$\mathcal{P}_{i,A|D} = \mathcal{P}_{i,AGID} \cup \mathcal{P}_{i,CNT}$$

where $\mathcal{P}_{i,AGID}$ is the set of propositional variables that denote the recipient of the message, and $\mathcal{P}_{i,CNT}$ is the set of proposition variables encoding dialogue action' content. With the number of agents is N, $|\mathcal{P}_{AGID}| = logN$. In most cases, $\mathcal{P}_{i,AGID}$ and $\mathcal{P}_{i,CNT}$ are of the same ontology for different agents T_i. The content propositions can be further decomposed as

$$\mathcal{P}_{i,CNT} = \mathcal{P}_{i,ID_W} \cup \mathcal{P}_V.$$

$\mathcal{P}_{i,ID_W}$ is the set of propositional variables that encode the identifier that denotes agent T_i's knowledge of the world state and its action variables. We denote the size of this set of variables as K_i where:

$$K_i = |\mathcal{P}_{i,ID_W}| = log|\mathcal{P}_{i,S|W} \cup \mathcal{P}_{i,A|W}|$$

$\mathcal{P}_V$ contains a variable v where $v = \{0, 1\}$ to indicate the truth value of variables with ID encoded by $\mathcal{P}_{ID,S|W} \cup \mathcal{P}_{ID,A|W}$ to communicated.

For notional convenience, we denote variables in $\mathcal{P}_{i,S|D}$ as $x_{i,j,k}$, $lx_{i,j,k}$ and $cx_{i,j,k}$.

- $x_{i,j,k}$ is T_i's information about T_j's kth state/action variable.
- $lx_{i,j,k}$ denotes the validity of $x_{i,j,k}$, so that $lx_{i,j,k} = 1$ means that $x_{i,j,k}$ is valid and $lx_{i,j,k} = 0$ means that $x_{i,j,k}$ is invalid.
- $cx_{i,j,k}$ is about whether T_i has communicated its kth state variable to agent T_j so that $cx_{i,j,k} = 1$ means that the value of T_i's kth variable has been communicated to T_j, $cx_{i,j,k} = 0$ means that the value of T_i's kth variable has not been communicated to T_j.

Notice that

$$|\mathcal{P}_{i,S|D}| = 3N * K + N$$

assuming that $K_i \approx K_j$, $i \neq j$, and writing this as K;

$$|\mathcal{P}_{i,A|D}| = logN + logK + 1.$$

In total, the number of propositional variables of $\mathcal{M}_{|D}$ is

$$(3N * K + N + logN + logK + 1) * N.$$

6.2 Action and communication

A communication $tell(i, j, x_k, v)$ will result in two state transitions. One will involve agent T_i and the other will involve T_j.

Definition 7 *For T_i, set $Pr_{a_{i|D}}(s_{i|D}, s'_{i|D}) = 1$ if $cx_{i,j,k} = 0$ and $\vec{y}_{i,agid} = j$, $\vec{y}_{i,ID_W} = k$, $cx'_{i,j,k} = 1$; otherwise $Pr_{a_{i|D}}(s_{i|D}, s'_{i|D}) = 0$. Correspondingly set the communication cost $C_{a_{i|D}}(s_{i|D}, s'_{i|D}) = 1$ if $Pr_{a_{i|D}}(s_{i|D}, s'_{i|D}) = \mathcal{C}$ where $\mathcal{C}$ is a constant of the cost of communicating a bit of information.*

This cost should be set relatively to the metric of the external world MDP model's reward measurement. Another choice is to have the reward function composed of a vector of rewards $\langle R_{|W}, R_{|D}\rangle$ — one for the external world and one for the dialogue — and then have the algorithms update each dimension using simple arithmetic operations according to the external world state transitions and the dialogue state transition accordingly; when comparison is needed, depending on the applications, different criteria can be applied in the comparison: e.g. the external world reward can take the precedence or discounting the communication costs, and etc.

Definition 8 *For T_j, set $Pr_{a_{i|D}}(s_{j|D}, s'_{j|D}) = 1$ if $lx_{i,j,k} = 0$ and $\vec{y}_{i,agid} = j$, $\vec{y}_{i,ID_W} = k$, then $lx_{i,j,k} = 1$, and in addition*

- *if $\vec{y}_{i,v} = 0$, then $x'_{j,i,k} = 0$;*
- *if $\vec{y}_{i,v} = 1$, then $x'_{j,i,k} = 1$;*

otherwise $Pr_{a_{i|D}}(s_{j|D}, s'_{j|D}) = 0$

We need to define an additional dialogue action $tell(i, j, z, v)$ which means agent T_i tells the agent T_j his is ready to act. It will incur two state transitions:

Definition 9 *For T_i, set $Pr_{a_{i|D}}(s_{i|D}, s'_{i|D}) = 1$ if $z_i = 0, z'_i = 1$; otherwise $Pr_{a_{i|D}}(s_{i|D}, s'_{i|D}) = 0$. Correspondingly set the communication cost $C_{a_{i|D}}(s_{i|D}, s'_{i|D}) = \mathcal{C}$ if $Pr_{a_{i|D}}(s_{i|D}, s'_{i|D}) = 1$ where $\mathcal{C}$ is the communication cost constant set by the application.*

Definition 10 *For T_j, set $Pr_{a_{i|D}}(s_{j|D}, s'_{j|D}) = 1$ if $z_i = 0$ in $s_{j|D}$, then set $z'_i = 1$; otherwise $Pr_{a_{i|D}}(s_{j|D}, s'_{j|D}) = 0$.*

6.3 Joint dialogue transitions

Now we consider the transitions that result from joint dialogue actions Recall that a dialogue is a sequence:

$$
\begin{array}{rcl}
a_{|D} & = & [\\
 & & \quad tell(1, j_1, k_1, v_1), \\
 & & \quad tell(2, j_2, k_2, v_2), \\
 & & \quad \vdots \\
 & & \quad tell(N, j_N, k_N, v_N) \\
 & & \qquad\qquad\qquad]
\end{array}
$$

In addition to the general behavior model, we impose further constraints on the probabilistic characterization Pr_i as follows:

- External world transitions:

  $$Pr_{\langle a_{i|W} \rangle}(\langle s_{i|W}, s_{i|D} \rangle, \langle s'_{i|W}, s'_{i|D} \rangle) = Pr_{a_{i|W}}(s_{i|W}, s'_{i|W})$$

  where the transition only happens when all the agents agree to act, namely the action bit z_j in $s_{i|D}$ is set to 1 for all $j = 1 \ldots N$ assuming there is a synchronization protocol to achieve these. Other approaches are possible, for example, in the joint probability action model we can specify the utility of the joint actions with some agents being idling, and having actions bits or timing information enabled in the dialogue states to embed the action synchronization problem into the utility model. We will leave this possibility for future research.

- Dialogue state transitions:

  $$Pr_{\langle a_{i|D} \rangle}(\langle s_{i|W}, s_{i|D} \rangle, \langle s_{i|W}, s'_{i|D} \rangle) = Pr_{a_{i|D}}(s_{i|D}, s'_{i|D})$$

  when the action bit z_j in $s_{i|D}$ is not set to 1 for all $j = 1 \ldots N$.

Definition 11 *For the joint dialogue state transition characterization, set* $Pr_{a_{|D}}(s_{|D}, s'_{|D}) = 1$*, if for every* $i = 1 \ldots N$

- $Pr_{a_{i|D}}(s_{i|D}, s'_{i|D}) = 1$*,*
- $Pr_{a_{j|D}}(s_{i|D}, s'_{i|D}) = 1$ *for all the* $j \neq i$

otherwise $Pr_{a_{|D}}(s_{|D}, s'_{|D}) = 0$*. Correspondingly the communication cost is*

$$C_{a_{|D}}(s_{|D}, s'_{|D}) = \Sigma_{i=1}^{N} C_{a_{i|D}}(s_{i|D}, s'_{i|D})$$

if $Pr_{a_{|D}}(s_{|D}, s'_{|D}) > 0$*.*

6.4 A specific version of $F_{i|W \to D}$ and $F_{i|D \to W}$

Having described our general model of team planning with communication, the last thing we do in this paper is to describe a specific implementation of the these functions.

Definition 12

$$F_{i|W \to D}(s_{i|W}) = \{s_{i|D}\}$$

where $s_{i|D}$ *is the agent* T_i*'s dialogue state with*

- $x_{i,i,k}$ *set to their corresponding values of its world state* $s_{i|W}$
- $lx_{i,i,k}$ *set to 1; and other* $lx_{i,j,k}$ *set to 0*
- *all* $cx_{i,j,k}$ *set to 0 (for* $j = 1 \ldots N$*)*

Definition 13

$$F_{i|W \to D}(a_{i|W}) = \{s_{i|D}\}$$

where $s_{i|D}$ *is the agent* T_i*'s dialogue state with*

- $x_{i,i,k}$ *set to their corresponding values of its world action* $a_{i|W}$
- $lx_{i,i,k}$ *set to 1; and other* $lx_{i,j,k}$ *set to 0*
- *all* $cx_{i,j,k}$ *set to 0 (for* $j = 1 \ldots N$*)*

Definition 14

$$F_{i|D \to W}(s_{i|D}) = \{s_{i|W}\}$$

where $s_{i|W}$ *is the agent* T_i*'s dialogue state with the bit values of the world state* $s_{i|W}$ *are set corresponding to those in* $s_{i|D}$*.*

Definition 15

$$F_{i|D \to W}(s_{i|D}) = \{a_{i|W}\}$$

where $a_{i|W}$ *is the agent* T_i*'s dialogue state with m the bit values of the world state* $a_{i|W}$ *are set corresponding to those in* $s_{i|D}$*.*

7 Related Work

The work we describe here differs from existing work on MDP models of multiagent systems in a number of ways. Perhaps most obviously, in the simplest model of multiagent MDPs [2], only actions are decomposible into individual agents' actions — states, state transition probabilities and rewards are specified for the whole system. Thus whereas our model reflects our focus on ad-hoc teams that are composed of independent elements that are brought together for a specific operation, the model in [2] considers the team to be the basic component and factors the individual agents' actions out from the joint policy.

In [1], the DEC-MDP model for external world is similar to our, but their communication model is just "yes" or "no" on whether individual agent should communicate the whole vector of local state variables to all the other agents. In their model, a myopic decision is made on whether to communicate or not, and only one-short or a k-step look ahead benefit of the communication is considered, and the approach doesn't revise the external world policy and its expected utilities when revising the communication policy. This is so partly because their model does not have as clean a concept as ours for how communication can affect the execution of a external world policy and in turn affect the expected utility of the policy.

Another model that is similar to the one we describe is the COM-MTDP (communicative multiagent team decision problem) [5]. COM-MTDP is multiagent teamwork model based on POMDP (partial observable Markov Decision Process) models. In the COM-MTDP model, the system's states can not be decomposed into individual agents' states, namely all the agents share the same set of joint states. The system's joint action is composed of individual agents' actions. Then the state transition probabilities and rewards are defined on the joint states and joint actions. From the same joint state, these agents can have different observations probabilistically, and collectively the agents can have joint observations.

The COM-MTDP model also includes a mental state — the belief state — component to bridge between the decision theoretic model and the BDI model. The dialogue state of our model provide some similar functions as the belief state component in the COM-MTDP model but the dialogue states and dialogue actions are modeled as another MDP in our system making it a more comprehensive and systematic way to design and model a team with communication structure in which the team can share their joint intentions — achieving the goal states from their initial states or maximizing their joint utilities — and maintain the execution of the courses of actions to achieve their joint intentions. Another difference is that our model explicitly enables a gradual scheme of communication by having the agents conduct the communication at the bit level of their states or observations. Finally, our model can be easily extended to be based on an underlying POMDP model.

More recently, along the road of COM-MTDP, in [7], a revised model based on networked distributed POMDP model is proposed. In the model, the joint states can be decomposed as ours. The model allows the agents to carry out their individual plans (projected from the join policy) for k steps, and then enter the communication phase to communicate the observation/action histories to revise their original planned joint POMDP policy. The communication scheme they proposed actually also lack long term view of how the communication can affect the external policy execution.

8 Conclusions

In this paper we have described an initial model for team planning with communication that extends our previous, symbolic, model to become a full decision-theoretic model. Such a model not only extends the representational capabilities of our previous work, but also the scope of solution concepts. Whereas before we could identify plans that might work, and plans that were guaranteed to work, solutions to the new model will also allow us to identify how likely particular plans are to succeed (by looking at trajectories through the state space) and to compute the expected utility of policies.

Future work will be directed towards the implementation of this approach and its testing on representative examples of team planning.

Acknowledgments

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] R. Becker, A. Carlin, V. Lesser, and S. Zilberstein. Analyzing Myopic Approaches for Multi-Agent Communication. *Computational Intelligence*, 25(1):31–50, February 2009.

[2] C. Boutilier. Planning, learning and coordination in multiagent decision processes. In *TARK '96: Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pages 195–210, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[3] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[4] D. Nau, M. Ghallab, and P. Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[5] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:2002, 2002.

[6] Y. Tang, T. Norman, and S. Parsons. A model for integrating dialogue and the execution of joint plans. In *Proceedings of the Eigth International Joint Conference on Autonomous Agents and Multiagent Systems*, Budapest, Hungary, May 10-15 2009.

[7] M. Tasaki, Y. Yabu, Y. Iwanari, M. Yokoo, M. Tambe, J. Marecki, and P. Varakantham. Introducing communication in dis-pomdps with locality of interaction. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, 2:169–175, 2008.