Agent-based Dialogues to Support Plan Execution by Human Teams

Yuqing Tang
PhD Program in Computer Science
Graduate Center,
City University of New York
365 Fifth Avenue
New York, NY 10016, USA
Email: ytang@gc.cuny.edu

Timothy J. Norman
Dept of Computing Science
The University of Aberdeen
Aberdeen, AB24 3UE, UK
Email: t.j.norman@abdn.ac.uk

Simon Parsons
Dept of Computer & Information Science
Brooklyn College,
City University of New York
2900 Bedford Avenue
Brooklyn, NY 11210 USA
Email: parsons@sci.brooklyn.cuny.edu

***Abstract*—Analysis of communications in human teams suggests that an important form of communication between team members is an "information providing" dialogue, in which team members update their fellows with information that they regard as important to the task at hand. In this paper we introduce and analyse a formal model of such a form of dialogue, seeing this as a necessary first step in providing software support for this kind of communication.**

I. INTRODUCTION

This paper deals with managing collaboration in a team. In particular, we are interested in teams engaged in military missions, and teams in which members may come from different parts of an international coalition. In such situations, effective coordination can be problematic, with units unable to communicate easily, and handicapped by having been trained to operate under rather different doctrines. It is our contention that, with careful design, software agents can support effective collaboration in teams, and can overcome some of the problems with coalition forces [1]. Extrapolating from existing applications of software agents:

- Agents can filter messages, preventing unnecessary messages from reaching specific human team members, and protecting them from distraction or information overload. [13]
- Agents can coordinate the activities of human team members [4], again reducing the cognitive burden on human operatives.
- Agents can ensure that relevant information is passed between human team members, facilitating timely delivery of crucial data. [20]
- Agents can help to enforce the correct protocol for team behavior, ensuring that human team members follow guidelines [8], [9].

For agents to be used in this way, they need to be programmed with some notion of what dialogues between human team members are to be expected, required, and allowed. A promising approach to specifying dialogues is the use of formal dialogue games [14], [19], and a number of authors have developed such systems [7], [16], [17], [18]. Particularly influential in this area is the work of Walton and Krabbe [24], who discuss six different types of dialogue that fit within the dialogue game framework:

1) **Information-Seeking Dialogues** One participant seeks the answer to some question(s) from another participant, who is believed by the first to know the answer(s);
2) **Inquiry Dialogues** Participants collaborate to answer some question or questions whose answers are not known to any one participant;
3) **Persuasion Dialogues** One party seeks to persuade another party to adopt a belief or point-of-view he or she does not currently hold. Persuasion dialogues begin with one party supporting a particular statement which the other party to the dialogue does not, and the first seeks to convince the second to adopt the proposition. The second party may not share this objective.
4) **Negotiation Dialogues** The participants bargain over the division of some scarce resource in a way acceptable to all, with each individual party aiming to maximize his or her share. The goal of the dialogue may be in conflict with the individual goals of each of the participants.[1]
5) **Deliberation Dialogues** Participants collaborate to decide what course of action to take in some situation. Participants share a responsibility to decide the course of action, and either share a common set of intentions or a willingness to discuss rationally whether they have shared intentions.
6) **Eristic Dialogues** Participants quarrel verbally as a substitute for physical fighting, with each aiming to win the exchange.

Walton and Krabbe allow for dialogues to be combinations of these different types, and they make no claims that this classification is complete. Girle, for example, discusses command dialogues [10] while Cogan *et al.* [6] describe a series of question-led dialogues that are distinct from Walton and Krabbe's information-seeking dialogue.

---

[1] Note that this definition of negotiation is that of Walton and Krabbe. Arguably negotiation dialogues may involve other issues besides the division of scarce resources.

In this paper, we formalise a new kind of dialogue that we have identified in the conversations of human members of teams that are engaged in military operations. This "information providing" dialogue type was identified following the analysis of transcripts from experiments conducted with human teams operating in simulated tactical military operations [23]. The form that this type of dialogue takes is one in which members update the rest of the team with new information that comes to light. Clearly, the mechanism of the dialogue itself is not complex — it just involves uttering the new information — but the important aspect is identifying when it is appropriate to make the utterances.

To reason about this, and to do it in a way that will be able to deal with teams operating in the real world, we have to go far beyond the kind of simple propositional language that was used in dialogue systems such as that of [16]. Instead we need the ability to reason about non-deterministic events, and the way that those events unfold over time. Borrowing from the language of AI planning systems, we start our formal system with a model of states and the transitions between states.

II. THE STATE TRANSITION MODEL

We use a state-space model, popular in the field of non-deterministic planning [12], as a basis for our formalisation. *States* are objects that capture some aspect of a system, and *actions* are transitions between states. States and actions together define a *state-space*. When actions are non-deterministic [2] then what one seeks for any state-space is a *policy*; i.e. a specification of which action one should take in every state. We define a non-deterministic domain to be a tuple $\mathcal{M} = \langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ where

- $\mathcal{P} = \mathcal{P}_S \cup \mathcal{P}_A$ is a finite set of propositions;
- $\mathcal{S} \subseteq 2^{\mathcal{P}_S}$ is the set of all possible states;
- $\mathcal{A} \subseteq 2^{\mathcal{P}_A}$ is the finite set of actions; and
- $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is the state-transition relation.

A propositional language $\mathcal{L}$ can be defined by allowing standard connectives $\wedge, \vee, \rightarrow, \neg$ over the propositions in $\mathcal{P}$. A state $s = \{p_1, p_2, \ldots, p_k\}$, $s \subseteq \mathcal{P}_S$, means that the propositions $p_1, p_2, \ldots, p_k$ are true in state s and all other propositions in $\mathcal{P}_S$ are false — we therefore assume some form of closed-world. In other words, each state s is encoded by a formula

$$\gamma = \bigwedge_{p_i \in s} p_i \wedge \bigwedge_{p_j \notin s \text{ and } s \in \mathcal{P}_S} \neg p_j$$

We denote that a formula γ is true in s by $s \models \gamma$. The set of states that satisfies formula $\gamma \in \mathcal{L}$ is denoted by S_γ, where $S_\gamma = \{s | s \models \gamma\}$.[2] Actions are encoded in a similar way to states. Action $a = \{p_1, p_2, \ldots, p_l\}$, $a \subseteq \mathcal{P}_A$ means that propositions $p_1, \ldots, p_l$ are true and all other formula in $\mathcal{P}^A$ are false. We denote that a formula α is true in a by $a \models \alpha$.

With states and actions defined, the state-transition relationship can then be specified by a set SR of triples: $SR = \{\langle \gamma, \alpha, \gamma' \rangle\}$ where $\gamma, \gamma' \in \mathcal{S}$ and $a \in \mathcal{A}$. Each triple $\langle \gamma, \alpha, \gamma' \rangle$ corresponds to a transition segment $R_{\langle \gamma, a, \gamma' \rangle} = \{\langle s, a, s' \rangle | s \models \gamma, a \models \alpha, s' \models \gamma\}$, and together:

$$\mathcal{R}_{SR} = \bigcup_{\langle \gamma, \alpha, \gamma' \rangle \in SR} R_{\langle \gamma, \alpha, \gamma' \rangle}$$

We take a *policy* to be a set of state-action pairs,

$$\pi = \{\langle s, a \rangle | s \in \mathcal{S} \text{ and } a \in \mathcal{A}(s)\}$$

where

$$\mathcal{A}(s) = \{a | \exists \langle s, a, s' \rangle \in \mathcal{R}\}$$

is the set of actions that are applicable in s. A policy can be specified by a set of pairs composed of a formula, $\gamma \in \mathcal{L}$, and an action, $a \in \mathcal{A}$: $SA = \{\langle \gamma, a \rangle\}$. Each pair $\langle \gamma, a \rangle$ corresponds to a policy segment: $\pi_{\langle \gamma, a \rangle} = \{\langle s, a \rangle | s \models \gamma\}$, and together

$$\pi_{SA} = \bigcup_{\langle \gamma, a \rangle \in SA} \pi_{\langle \gamma, a \rangle}$$

The space of all policies is denoted by Π. The set of states in a policy π is $S_\pi = \{s | \langle s, a \rangle \in \pi\}$. An execution structure induced by the policy π is a directed graph $\Sigma_\pi = (V_\pi, E_\pi)$ where

$$V_\pi = S_\pi \cup \{s | \langle s_p, a, s \rangle \in \mathcal{R}, s_p \in V_\pi \text{ and } \langle s_p, a \rangle \in \pi\}$$

is the set of nodes of Σ_π, which represent all possible states in $\mathcal{M}$ that can be generated by executing the actions in π. Typically this is a strict subset of the full state space.

$$E_\pi = \{\langle s, s' \rangle | \langle s, a, s' \rangle \in \mathcal{R} \text{ and } \langle s, a \rangle \in \pi\}$$

is the set of arcs between the nodes of Σ_π which represent possible transitions caused by the actions in π.

To describe the behavior of a team, we need to prescribe more structure over the actions. We assume that there is a set of n individuals labeled by $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ in the system. We will call these individuals *agents*, not distinguishing whether these are software agents or humans. We call the actions in the set $\mathcal{A}$ *joint actions* of these agents. Each action $a \in \mathcal{A}$ is a tuple of actions of individual agents, so $a = [a_1, \ldots, a_n]$. That is each action $a \in \mathcal{A}$ can be further decomposed into n actions $a_i \in \mathcal{A}_i$ of individual agents T_i. Each $\mathcal{A}_i$ is defined to be a subset of the propositions in $\mathcal{P}_{\mathcal{A}}$. By overloading the notion, we also denote $a \models a_i$ if agent T_i's action is a_i in a joint action a. In total, we have:

$$\mathcal{A} = \prod_i \mathcal{A}_i$$

In addition to this notion of joint action, there is a joint action version of each individual agent's state transition relation and policy. Thus, agent T_i, with state transition relationship R_i, has a corresponding joint action state transition relation:

$$R_i^* = \{\langle s, a, s' \rangle | \langle s, a_i, s' \rangle \in R_i \text{ and } a \models a_i\rangle\}$$

The joint action policy for agent T_i is defined as:

$$\pi_i^* = \{\langle s, a \rangle | \langle s, a_i \rangle \in \pi \text{ and } a \models a_i\rangle\}$$

---

[2] Note that $S_{p_1 \wedge p_2 \wedge \ldots \wedge p_k} \neq \{s\}$ where $s = \{p_1, p_2, \ldots, p_k\}$ because S_γ doesn't make the closed world assumption; that is, we assume that the unspecified propositions are false.

The induced joint state transition relationship is then:

$$\mathcal{R} = \bigcap_i R_i^*$$

And the induced joint policy is:

$$\pi = \bigcap_i \pi_i^*$$

An additional formula $\beta \in \mathcal{L}$ can be introduced to constrain possible combinations so that $\mathcal{A}(\beta) = \{a \in \mathcal{A} | a \models \beta\}$. For example,

$$\beta = \bigwedge_{i=1}^{n} \bigwedge_{j \neq i} a_i \rightarrow \tau_j$$

where τ_j is a special symbol for an empty action. This means that no concurrent actions are allowed, so that all the agents have to take actions in turn. The corresponding constrained joint state transition relationship is:

$$\mathcal{R}_i^*(\beta) = \{\langle s, a, s' \rangle | \langle s, a, s' \rangle \in \mathcal{R}_i \text{ and } a \models \beta \rangle\}$$

and the corresponding constrained joint policy is:

$$\pi_i^*(\beta) = \{\langle s, a, s' \rangle | \langle s, a, s' \rangle \in \pi_i \text{ and } a \models \beta \rangle\}$$

We will use the above state-transition model to represent both the dialogue mechanism and the topics of the dialogue. At the level of the topic of a dialogue, the state transition model encodes the agents' information about the external world and what they want to do in the external world (that is what policy they have). At the level of the dialogue model, the state transition model encodes the dialogue mechanism that these agents are using, and so regulates the dialogue.

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 $\mathcal{S}_{|D}$ 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 $\mathcal{S}_{|W}$ the external world states. However, when the state transition model is obvious from the context, we will omit the subscripts.

The use of state transition systems to model both the dialogue mechanism and the external dynamics is intended to utilize recent advances in the area of AI planning. These advances, especially in hierarchical task network non-deterministic planning (for example [12]), will aid in the development of systems that can handle complex dialogue behavior and help in the definition of broad solution concepts for dialogue. For example, the concept of a strong solution can be borrowed to identify when a dialogue mechanism will guarantee success, and the concept of a weak solution can be borrowed to identify when a dialogue mechanism will guarantee that success is possible. We will discuss this more below.

III. A DIALOGUE MODEL

Having established a language that is sufficiently rich to describe agent plans and actions, we can construct a dialogue model that uses this language.

A. *The general model*

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 $\mathcal{M}_{i|W} = \langle \mathcal{P}_{i|W}, \mathcal{S}_{i|W}, \mathcal{A}_{i|W}, \mathcal{R}_{i|W} \rangle$ 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 $\mathcal{M}_{|D} = \langle \mathcal{P}_{|D}, \mathcal{S}_{|D}, \mathcal{A}_{|D}, \mathcal{R}_{|D} \rangle$ for which there is a policy for conducting dialogues $\pi_{|D}$. 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. How these dialogues change the information state will be specified by the dialogue state transition relationship of these dialogue acts: $\mathcal{R}_{|D} \subseteq \mathcal{S}_{|D} \times \mathcal{A}_{|D} \times \mathcal{S}_{|D}$. Depending on the specific dialogue, we may distinguish a set of initial dialogue states $I_{|D} \subseteq \mathcal{S}_{|D}$ and a set of goal dialogue states $G_{|D} \subseteq \mathcal{S}_{|D}$ (see [21] for an example).

As is usual for state-transition models [2], a policy for a dialogue $\pi_{|D} = \{\langle s_{|D}, a_{|D} \rangle\}$ specifies what dialogue action should be taken in any given dialogue state to reach the goal states $G_{|D}$ from the initial states $I_{|D}$ at the least expected cost. 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*. The effects of conversation policies on a dialogue model $\mathcal{M}_{|D}$ can be summarized by the triple $\langle I_{|D}, \pi_{|D}, G_{|D} \rangle$ that states which initial and goal states are related by the policy.

Two conversation policies can be combined together to specify more complicated dialogues addressing different set of goals. There are two ways to combine two conversation policies π_1 and π_2:

- Union. The union of π_1 and π_2:

  $$\pi_1 \cup \pi_2 = \{\langle s, a \rangle | \langle s, a \rangle \in \pi_1 \text{ or } \langle s, a \rangle \in \pi_2\}$$

  extends the allowed behaviors of π_1 to include the behaviors of π_2. The corresponding effect of such a policy is $\langle I_{1|D} \cup I_{2|D}, \pi_1 \cup \pi_2, G_{1|D} \cup G_{2|D} \rangle$
- Join. The production, or join, of π_1 and π_2

  $$\pi_1 \cap \pi_2 = \{\langle s, a \rangle | \langle s, a \rangle \in \pi_1 \text{ and } \langle s, a \rangle \in \pi_2\}$$

  constrains the allowed behaviors of π_1 to only contain the behaviors of π_2. Here the corresponding effect is $\langle I_{1|D} \cap I_{2|D}, \pi_1 \cap \pi_2, G_{1|D} \cap G_{2|D} \rangle$

With this general model of dialogue in mind, we can specify information providing dialogues.

B. *Information-providing dialogues*

An information-providing dialogue is a dialogue in which one agent pushes (in the same sense as *push technology* [11]) information to its teammates because it believes that the information being pushed is helpful to its teammates in executing their policies.

We assume that, as in the general model, there is a set of n agents $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$, and each agent T_i has a world model $\mathcal{M}_{i|W} = \langle \mathcal{P}_{i|W}, \mathcal{S}_{i|W}, \mathcal{A}_{i|W}, \mathcal{R}_{i|W} \rangle$ and a policy over that world model $\pi_{i|W} = \{\langle s_i, a_i \rangle\}$. As shown in Section II, the corresponding induced joint state transition model is $\mathcal{M}_{|W} = \langle \mathcal{P}_{|W}, \mathcal{S}_{|W}, \mathcal{A}_{|W}, \mathcal{R}^*_{|W} \rangle$ where

$$\mathcal{R}^*_{|W} = \bigcap_i \mathcal{R}^*_{i|W}$$

The induced policy over this joint state model is

$$\pi_{|w} = \bigcap_i \pi^*_{i|W}$$

The agents can use these models to reason about the behaviors of one another in the real world.

In addition, we assume that each agent T_i holds a belief state $s_{i|D} = \langle s_{i|W}, s'_{i|W} \rangle$ which is composed of its perception of the previous world state $s_{i|W}$ and the current world state $s'_{i|W}$. The agent's belief model includes the relationship between all feasible states of the world, along with a suitable measure of belief in the transition. In other words, for every pair of states $s_{i|W}$ and $s'_{i|W}$ between which the agent can move there is an $\mathcal{R}_{i|W}(s_{i|W}, a_i, s'_{i|W})$ for some $a_i \in \mathcal{A}_i$.

Finally, we assume that the set of policies $\{\pi_{i|W}\}$ is known to every agent — so that every agent knows every policy of every agent — or equivalently we assume that there is a dialogue mechanism which can be used by an agent to retrieve the policies of any other agent efficiently. This may be an information-seeking dialogue, in the terminology of Walton and Krabbe [24] if agents can compute policies on their own, or a deliberation dialogue if the agents must collaborate to come up with the policies.

Under these assumptions, we can define the dialogue mechanism formally below by specifying the set of dialogue states, the set of dialogue actions, the basic dialogue model as a state transition relationship, and the application dependent dialogue model as a set of individual dialogue policies over the dialogue state transition model. We start with the model of dialogue state and action:

Definition 1 (Dialogue state): A *dialogue information state* of an agent T_i is composed of the belief of previous world state $s_{|W}$ and the current world state $s'_{|W}$, together $s_{i|D} = \langle s_{i|W}, s'_{i|W} \rangle$. The joint dialogue state is $s_{|D} = [s_{i|D}]_{i=1,\ldots,n}$. The corresponding set of propositions for the dialogue is $\mathcal{P}_{|D} = \mathcal{P}_{|W} \cup \mathcal{P}'_{|W}$ where $\mathcal{P}'_{|W}$ is a set of propositions copying $\mathcal{P}_{|W}$ with the same meaning but with different symbols so that the same logical structures are defined over the previous states and the current states.

Definition 2 (Dialogue action): A dialogue action is of the form $tell_{i\rightarrow j}(p)$ where $p \in \mathcal{P}$ and T_i and T_j are two agents participating in the dialogue. The meaning of $tell_{i\rightarrow j}(p)$ is that agent T_i tells T_j that a proposition $p \in \mathcal{S}_{|W}$ is true in its current belief state $s'_{|W}$, namely $s'_{|W} \models p$. The set of actions available to agent T_i is $\mathcal{A}_i = \{tell_{i\rightarrow j}(p) | p \in \mathcal{P}$ and $j = 1, \ldots, n\} \cup \{\tau\}$ where τ is an idle action. The set of joint dialogue actions is $\mathcal{A} = \prod_i \mathcal{A}_i$.

The restriction of the content of $tell$ to be a proposition p can be relaxed so that it is a sentence in $\mathcal{L}$ without affecting any of the definitions here, but we restrain from doing so to keep the model simple for the purposes of this paper.

We are developing this dialogue system as a step towards implementing a machine dialogue that improves the collaboration of a human team, and so we assume that it takes negligible time to execute a dialogue action since carrying out such an action takes much less time than executing a real world action (usually carried out by a human). If the execution time of the dialogue actions does matter, we can use a concurrent state transition model which combines the dialogue state transition model and external world state transition model; examining models that can handle the dialogue and the external world actions simultaneously is part of our future research.

We can also study a simpler version of the current model where we restrict the participants of a dialogue to take turns; i.e. in a joint action, exactly one agent is allowed to perform a $tell$ action, and all other agents can only perform the idle action τ. As shown in Section II, this can be done by introducing the following constraint where τ_j is a special symbol for the idle action, or noop, for agent j.

$$\beta = \bigwedge_{i=1}^{n} \bigwedge_{j \neq i} a_i \rightarrow \tau_j$$

We take the state transition model of the whole dialogue to be the product of the state transition models of individual agents, and we compose these models to get a joint model of the dialogue:

Definition 3 (Individual dialogue state transition): There are two *information state transitions* at the level of individual agents associated with the utterance $a_i = tell_{i\rightarrow j}(p)$ by agent T_i: the transitions for agents j (the hearer) and i (the speaker) respectively.

$$\begin{array}{lll} \exists r \in R_j & \text{s.t.} & r \doteq \langle\langle s, s'\rangle, tell_{i\rightarrow j}(p), \langle ss, ss'\rangle\rangle \in \mathcal{R}_{j|D} \\ \exists rr \in R_i & \text{s.t.} & rr \doteq \langle\langle s, s'\rangle, tell_{i\rightarrow j}(p), \langle s, s'\rangle\rangle \in \mathcal{R}_{i|D} \end{array}$$

where $ss = s$ and $ss' = s' \cup \{p\}$.
In the above definition, $r \in R_j$ specifies that the belief of agent T_j, the listener, is updated by adding the proposition p into its current set of beliefs about the world. Similarly $rr \in R_i$ specifies that the belief of T_i, the speaker, is not changed by the dialogue act. These updates are the equivalent of the commitment store updates in [16], [17].

Definition 4 (Dialogue joint state transition): The *joint dialogue state transition relationship* $\mathcal{R}$ associated with $a_i = tell_{i\rightarrow j}(p)$ is defined as follows

$$\mathcal{R} = \bigcap_{T_i \in \mathcal{T}} \mathcal{R}^*_i$$

This is equivalent to

$$\begin{array}{rcl} \mathcal{R} & = & \{\langle s, a, s'\rangle | s \models s_i, s' \models s'_i, a \models a_i \text{ such that} \\ & & \langle s_i, a_i, s'_i \rangle \in \mathcal{R}_i \text{ for all } i = 1, 2, \ldots, n\} \end{array}$$

This characterizes how a dialogue action, or utterance, leads to an update in belief of all the agents in the system, and how the effect is composed from individual views of the changes. After defining the dialogue model, we need to specify the initial state of the dialogue. We assume that at the beginning of a dialogue the belief states of all the agents are empty. There is a bootstrap mechanism modeled in these agents' external world state transitions at the beginning.

Definition 5 (Dialogue initial state): The initial state of a dialogue is $I = [\langle s_i, s_i' \rangle]_{i=1,\dots,n}$ with $\langle s_i, s_i' \rangle = \langle \emptyset, \emptyset \rangle$ for every agent T_i.

The goal of an information providing dialogue is to push the information about the state of world which will affect the other agents' execution of their policies. There can be many concepts of what information will affect the other agents, or what information is relevant to others. For example, one might be information that distinguishes two states in S_π (an agent's policy states) affecting that agent's choice of actions. Another example might be the information defining a state in S_π whose effect is to cause a specific action to be taken. For this paper, we focus on the relevance of information to the policies of other agents.

Definition 6: The goal state of the dialogue is $G = \{[\langle s_i, s_i' \rangle]_{i=1,\dots,n}\}$ where every $\langle s_i, s_i' \rangle$ satisfies that, for every agent T_i with an external policy $\pi_{i|W}$, $s_i \models p$ whenever

- there is an agent T_j with $s_j' \models p$ and $s_j \not\models p$, and
- p satisfies the criterion of being relevant to T_i's policy execution. As an example, we will employ the following: p is relevant to T_i's policy execution if there exists some $(s_i, a_i) \in \pi_{i|W}$ such that $s \vdash p$.

The above definition says that the agent should be able to have the belief that p is true in the state whenever (i) some other agent observes p to be true in the previous world state, and (ii) the truth of p will affect the agent's policy execution. In the example below, it is knowing p that will help agent T_i fire an action a.

Now we are ready to specify the information providing dialogue — a dialogue that specializes in pushing information.

Definition 7 (Information providing dialogue): An *information providing dialogue* is a mechanism in which each agent T_i is equipped with a conversation policy $\pi_{i|D} = \{\langle s_{i|D}, a_{i|D} \rangle\}$ where

- $s_{i|D} = \langle s_{i|W}, s_{i|W}' \rangle$ where $s_{i|W}$ denotes the previous world state and $s_{i|W}'$ denotes the current world state
- $a_i \in tell_{i \to j}(p)$ whenever
  - $p \notin s_{i|W}$ but $p \in s_{i|W}'$,
  - there is some $\langle s_{j|W}, a_{j|W} \rangle \in \pi_{j|W}$ such that $p \in s_{j|W}$

The above dialogue mechanism is just a straight forward translation of the goal into a dialogue policy — more complex conversation policies can also be defined. For ease of explanation, we just use this simple policy in this paper.

We can view the policy of the whole dialogue as a joint policy induced from the individual agents' policies.

Definition 8: The *joint dialogue policy* for a set of agents T_i is

$$\begin{aligned} \pi_{|D} \quad = \quad & \big\{\langle [s_{i|D}]_{i=1,\dots n}, [a_{i|D}]_{i=1,\dots n} \rangle \mid \\ & \langle s_{i|D}, a_{i|D} \rangle \in \pi_{i|D}, i = 1, \dots, n \big\} \end{aligned}$$

With these definitions, we can easily show some basic properties of our dialogue model.

Proposition 1: The information-providing dialogue will end with the dialogue in the goal state.

Proof: Trivial — it follows because the dialogue policy is just a straightforward translation of the dialogue goal. ■

Proposition 2: The execution complexity of a step for the dialogue policy $\pi_{i|D}$ is $2N + NM$ where $N = |\mathcal{P}|$ and $M = |\pi_{j|W}|$.

Proof: M is the size of the equivalent formula pair representation if the representation introduced in Section II is employed. The test of whether $p \in \xi(s_{i|W}')$ but not $p \in \xi(s_{i|W})$ can be done in at most $2N$ steps by scanning two list of propositions, and the test of whether $s \in S_{\pi_{j|W}}$ can be done in at most $N \cdot M$. The total execution time, then is $2N + NM$. ■

If the state representations are encoded as binary decision diagrams [3], it should be possible to find a more efficient decision procedure.

IV. EXAMPLE DIALOGUES

To demonstrate how these information-providing dialogues work, we will give two examples:

1) An information providing dialogue that makes use of information about locations; and
2) An information providing dialogue that includes location and time information.

These examples also demonstrate how the state transition modeling dialogues can be incrementally detailed to fit an application.

A. A location-based dialogue

In our first example, we abstract away the details of the external world by keeping only location information in the in state space and leave the other aspects of the state, world actions, and state transitions unspecified. There are two types of symbols in $\mathcal{P}_{|W}$, they are e_l and $t_{i,l}$ meaning, respectively, that an enemy is in location l and agent T_i is in location l. We assume that once an enemy has been discovered in a particular location it will never move.[3] We also assume that there is an external system (for example, an perception system) that will update information about the locations of agents in the world as they move (i.e. delete $t_{i,l}$ and add $t_{i,l'}$ if agent T_i moves from location l to location l') and the locations of enemies when they are discovered (i.e. add facts such as e_7). The goal of the dialogue is that if an agent finds that an enemy appears in a location where there is some other agent that has a policy

---

[3] Clearly unrealistic, and easy enough to overcome, this allows us to write down examples that are suitably short for this paper.

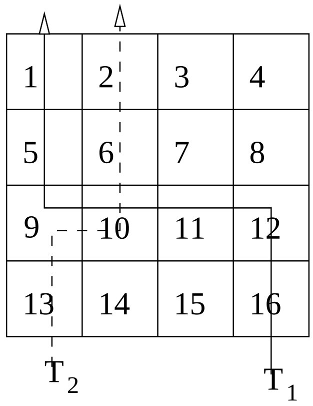

Fig. 1. The external world for the example

to pass through that location, then the first agent will notify his teammate of the presence of the enemy.

Definition 9: A *location-based information providing dialogue* is a mechanism in which each agent T_i is equipped with a conversation policy $\pi_{i|D} = \{\langle s_{i|D}, a_{i|D}\rangle\}$.

- $s_{i|D} = \langle s_{i|W}, s'_{i|W}\rangle$ where $s_{i|W}$ denotes the previous world state and $s'_{i|W}$ denotes the current world state.
- $a_i \in tell_{i\to j}(p)$ whenever
  - $e_l \notin s_{i|W}$ but $e_l \in s'_{i|W}$,
  - there is a $\langle s_{j|W}, a_{j|W}\rangle \in \pi_{j|W}$ such that $e_l \in s_{j|W}$

Example 1: Consider the following, very simple, example. The world is modeled as a 4 by 4 grid as in Figure 1. T_1's initial position is 16, and T_2's initial position is 13. T_1's goal position is 1 and T_2's goal state is 2. At any point in time an agent has two choices of action:

1) It can make a $move$ in a direction that is up, $down$, $left$ or $right$, this has the effect of moving the agent into the relevant location if it is vacant. If an enemy is in the relevant location, attempting to $move$ into it will cause damage to the agent, and the agent will not move, but the agent will learn the location of the enemy.
2) It can make a $defenceMove$ in any of the four directions. This represents a deployment in a defensive formation, which is costlier than a $move$ but makes it possible to move into the relevant location even if an enemy is deployed there, and prevents that enemy from doing any damage.

Agents can only detect the presence of an enemy in a particular location if they attempt to move into the same location, and we know that at the start of the scenario, there are no enemies in squares 13 and 16.

Clearly an agent should move towards its goal, using $defenceMove$ if it finds that a location it wants to move through is the location of an enemy. Two possible policies for T_1 and T_2 are presented in Figures 2 and 3 respectively.

These two policies are both complete if the effects of the agents' actions are deterministic, and both are partial if the actions are non-deterministic. For simplicity of explanation, we assume the world to be deterministic for the moment.

$$
\begin{array}{ll}
SA_{1|W} = \{ & \langle t_{1,16} \wedge \neg e_{12}, move(up)\rangle \\
 & \langle t_{1,16} \wedge e_{12}, defenceMove(up)\rangle \\
 & \langle t_{1,12} \wedge \neg e_{11}, move(left)\rangle \\
 & \langle t_{1,12} \wedge e_{11}, defenceMove(left)\rangle \\
 & \langle t_{1,11} \wedge \neg e_{10}, move(left)\rangle \\
 & \langle t_{1,11} \wedge e_{10}, defenceMove(left)\rangle \\
 & \langle t_{1,10} \wedge \neg e_{9}, move(left)\rangle \\
 & \langle t_{1,10} \wedge e_{9}, defenceMove(left)\rangle \\
 & \langle t_{1,5} \wedge \neg e_{5}, move(up)\rangle \\
 & \langle t_{1,5} \wedge e_{5}, defenceMove(up)\rangle \qquad \}
\end{array}
$$

Fig. 2. Example policies for T_1

$$
\begin{array}{ll}
SA_{2|W} = \{ & \langle t_{2,9} \wedge \neg e_{9}, move(up)\rangle \\
 & \langle t_{2,9} \wedge e_{9}, defenceMove(up)\rangle \\
 & \langle t_{2,10} \wedge \neg e_{10}, move(right))\rangle \\
 & \langle t_{2,10} \wedge e_{10}, defenceMove(right)\rangle \\
 & \langle t_{2,6} \wedge \neg e_{6}, move(up)\rangle \\
 & \langle t_{2,6} \wedge e_{6}, defenceMove(up)\rangle \qquad \}
\end{array}
$$

Fig. 3. Example policies for T_2

Using the location-based information-providing dialogue policy above, if agent T_1 encounters enemies at location 9 or 10, then it will tell agent T_2 so that the latter changes the actions it will take when moving into these two locations from $move$ to $defenceMove$, reducing the cost to the team of completing achieving their goals.

B. A time-sensitive location-based dialogue

We can extend the above example to take account of information about time. To capture this temporal information, we extend the set $\mathcal{P}_{|W}$ of propositional symbols in the previous dialogue with an additional type of symbol that represents the timing information corresponding to the location information of the other agents on the team and the enemies. These extended symbols are of the form $\omega = [\omega_1, \omega_2, \omega_3, \omega_4]$ is a vector of boolean propositions encoding the number from 1 to 16 to represent a discrete time frame. Each proposition $t_{i,l}$ is associated with a timing vector $\omega_{i,l}$, and each proposition e_l is associated with a timing vector ω_l to represent the time when the truth of the proposition is discovered.

The goal of the dialogue is that if an agent finds that enemy appears in a location where there is some other agent with an acting policy to pass through this location in the future, then the agent should notify its teammate.

Definition 10: A *timed location-based information providing dialogue* is a mechanism in which each agent T_i is equipped with a conversation policy $\pi_{i|D} = \{\langle s_{i|D}, a_{i|D}\rangle\}$ where

- $s_{i|D} = \langle s_{i|W}, s'_{i|W}\rangle$ where $s_{i|W}$ denotes the previous world state and $s'_{i|W}$ denotes the current world state.
- $a_i \in tell_{i\to j}(p)$ whenever
  - $e_l \notin s_{i|W}$ but $e_l \in s'_{i|W}$,

- there is a $\langle s_{j|W}, a_{j|W} \rangle \in \pi_{j|W}$ such that $s_{j|W} \models e_l \wedge t_{j,l}$ and $\omega_l - \omega_{j,l} \leq c$ where c is a constant representing time sensitive range.

Thus it is easy to specify what information about enemy location should be passed, and, clearly this form of dialogue will never pass more information than the purely location-based dialogue. For example, if T_1 encounters an enemy at location 10, the purely location-based approach would pass this information to T_2, unnecessarily, since, if the agents move at the same speed, T_2 would have already passed this point. However, the time-sensitive dialogue will not pass the information. Of course, for this to be correct, and save T_2 from unnecessary damage, the temporal information must be correct — it must take into account that T_2 may have encountered an enemy at location 9 and so still be on its way to 10.

In the above, the computation of $\omega_l - \omega_{i,l} \leq c$ can be substituted by any boolean functions over the propositions which computed the less-than-or-equal-to relationship over the integer encoding is being used.

The goal state of the dialogue is then that $G = \{[\langle s_i, s'_i \rangle]_{i=1,\ldots,n}\}$ satisfies that for every agent T_i with a policy $\pi_{i|W}$

if $s'_j \models t_{i,l} \wedge \omega_{i,l}$ and $(s_i, a_i) \in \pi_{i|W}$ then $s_i \models e_l \wedge \omega_l$ where $\omega_l - \omega_{i,l} \leq c$.

In other words, every agent knows the information needed for its policy if the information is available from any other agent.

V. CONCLUSIONS AND FUTURE WORK

This paper has made two main contributions. The first is to formalise dialogue mechanisms using a general state-transition model. State-models of dialogue protocols are not new, but we are the first to have used such a rich representation, both to describe the dialogue itself, and to describe the subject matter of the dialogue (which is typically taken to be a simple propositional language). This rich representation not only extends the subject matter of dialogues to include the kinds of objects that will need to be manipulated by agents supporting coalition teams, but also points to an implementation, as described below.

The second contribution of the paper is to provide a formalisation of some simple information-providing dialogues, a form of dialogue that is both new to work in formal dialogue models, and, from the analysis of the conversation of human teams, seems likely to be useful in practice.

Our future work is to move towards a practical implementation of this work. The conversation policy for information providing dialogues is a modular policy in the framework of [22], and the other policies outlined in [22] can be described in the same state-transition formalism that was introduced here. Once we have such descriptions for a set of conversation policies — and this formalisation is what we are working on now — we can move to an implementation that uses the same model-checking mechanisms that are used in modern AI planners (which, as discussed above, use the same representation that we have adopted here).

The key to using a state transition model for planning in a non-deterministic domain is to encode sets of states, sets of state transitions, sets of state-action pairs and the operations on these sets compactly with Quantified Boolean Formulae (QBF), and use binary decision diagrams [3] to represent and manipulate the QBF formulae efficiently. In this way, we can represent and manipulate sets of states, sets of state transitions and policies simultaneously instead of explicitly enumerating all the states and state transitions involved. Cimatti et. al. [5] provides a excellent description of using BDDs for planning in a non-deterministic domain and the corresponding solution concepts. Kuter and colleagues [12] combine the BDD approach with their own hierarchical planning network approach [15] to give a more natural way to characterize state transition models. Exactly the same techniques will provide us with the means to handle more complex dialogues than the one presented here, while also (in the other state-transition model) provide us with the ability to handle complex dialogue topics.

ACKNOWLEDGMENT

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] J. Allen, A. Mowshowitz, T. J. Norman, S. Parsons, and A. Preece. Managing collaboration in hybrid-agent teams. In *Proceedings of the First Annual Conference of the ITA*, University of Maryland, 2007.

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

[3] R. E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[4] H. Chalupsky, Y. Gil, C. A. Knoblock, K. Lerman, J. Oh, D. V. Pynadath, T. A. Russ, and M. Tambe. Electric elves: Agent technology for supporting human organizations. *Artificial Intelligence*, 23(2):11–24, 2002.

[5] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2):35–84, 2003.

[6] E. Cogan, S. Parsons, and P. McBurney. What kind of argument are we going to have today? In *Proceedings of the 4th International Conference on Autonomous Agents and Multi-Agent Systems*, New York, NY, 2005. ACM Press.

[7] F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In C. Castelfranchi and Y. Lespérance, editors, *Seventh Workshop on Agent Theories, Architectures, and Languages*, pages 141–156, Boston, USA, 2000.

[8] M. Esteva, D. Cruz, and C. Sierra. Islander: an electronic institution editor. In *Proceedings of the 1st International Conference on Autonomous Agents and Multi-Agent Systems*, New York, NY, 2002. ACM Press.

[9] M. Esteva, J. A. Rodriguez, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specification of electronic institutions. In *Agent-mediated Electronic Commerce*, number 1991 in Lecture Notes in Artificial Intelligence, pages 126–147. Springer Verlag, Berlin, Germany, 2001.

[10] R. Girle. Commands in Dialogue Logic. In D. M. Gabbay and H. J. Ohlbach, editors, *Practical Reasoning: Proceedings of the First International Conference on Formal and Applied Practical Reasoning (FAPR 1996), Bonn, Germany*, number 1085 in Lecture Notes in Artificial Intelligence, pages 246–260. Springer Verlag, Berlin, Germany, 1996.

[11] K. Kelly and G. Wolf. Push! *Wired*, 5.03, March 1997.

[12] U. Kuter, D. S. Nau, M. Pistore, and P. Traverso. A hierarchical task-network planner based on symbolic model checking. In *Proceedings of the International Conference on Artificial Intelligence Planning Systems*, 2005.

[13] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):31–40, 1994.

[14] P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language, and Information*, 11(3):315–334, 2002.

[15] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. Shop2: An htn planning system. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.

[16] S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *Proceedings of the 1st International Conference on Autonomous Agents and Multi-Agent Systems*, New York, NY, 2002. ACM Press.

[17] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[18] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.

[19] I. Rahwan, S. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and E. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, December 2003.

[20] S. D. Ramchurn, B. Deitch, M. K. Thompson, D. C. de Roure, N. R. Jennings, and M. Luck. Minimising intrusiveness in pervasive computing environments using multi-agent negotiation. In *Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems*, Boston, MA, 2004.

[21] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiations. In M. P. Singh, A. Rao, and M. J. Wooldridge, editors, *Intelligent Agents IV*, pages 177–192. Springer Verlag, Berlin, Germany, 1998.

[22] Y. Tang and S. Parsons. A dialogue mechanism for public argumentation using conversation policies. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems*, Estoril, Portugal, 2008.

[23] Y. Tang, S. Poltrock, H. Bowyer, and S. Parsons. Analysis of a recorded team coordination dialogues. In *Proceedings of the Second Annual Conference of the ITA*, Imperial College, London, 2008.

[24] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Series in Logic and Language. State University of New York Press, Albany, NY, 1995.