Argumentation-based dialogues for deliberation

Yuqing Tang and Simon Parsons
Department of Computer Science
Graduate Center
City University of New York
365 5th Avenue, New York
NY 10016, USA
ytang@gc.cuny.edu,parsons@sci.brooklyn.cuny.edu

ABSTRACT

This paper presents an argumentation-based approach to deliberation, the process by which two or more agents reach a consensus on a course of action. The kind of deliberation we are interested in combines both the selection of an overall goal, the reduction of this goal into sub-goals, and the formation of a plan to achieve the overall goal. We develop a mechanism for doing this and then proceed to describe how it can be integrated into a system of argumentation to provide a sound and complete deliberation system, before showing how the same process can be achieved through a multi-agent dialogue.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence and co-ordination; multiagent systems.*

General Terms

Languages, theory.

Keywords

Agent communication, dialogue games, argumentation.

1. INTRODUCTION

Multi-agent planning is clearly an important topic for the field of multi-agents systems, and, as one might imagine, has been widely studied and for a long time. As [4] points out, there is a large variety of approaches, from distributed versions of classical AI planning techniques like NOAH [3] and partial planning [6], to techniques that were developed to exploit specific attributes of multi-agent systems like joint intentions [14, 23], or the *intention-that* of SharedPlans [9]. Some of these approaches deal with multi-agent plans holistically [10], while others build plans for individual agents and then merge them [7]. Approaches as disperate as model checking [24] and auctions [26] have been adapted to generate multiagent plans.

In this paper we bring together aspects of multiagent planning and work in a field that has grown up more recently, argumentation-based dialogue [18]. While there has been much work on argumentation-based dialogue in the last few years—including that of Kraus [13], Maudet [15], McBurney [16], Reed [20], Schroeder *et al.* [21] and Sycara [22]—there is not yet a definitive account of what Walton and Krabbe [25] call *deliberation* dialogues. These are dialogues in which two or more agents converse to formulate a joint course of action.

At the time of writing, we have team formation dialogues Dignum *et al.* [5], dialogues about what should be done [8], dialogues in which one agent proposes a plan and then persaudes others to adopt it [17], and even a general purpose framework for deliberation [12]. However, our goal in this paper is to provide a form of dialogue which allows agents to exchange arguments about the details of means-ends planning. This is something we believe is essential if agents are going to rationally discuss what plans to adopt. In particular, we aim to develop a dialogue in which agents not only decide what to do, but create a plan *jointly*, with different sub-plans being suggested by different agents which then merge them to create an overall plan that they all agree on.

2. NOTATION

As usual when considering planning, whether the classical planning of STRIPS or the decision theoretic planning of POMDPS [1] we abstract the physical world into states, actions, and state transitions caused by the actions. States and actions are the basic objects in L, the underlying language used in our approach. The kind of procedure we are interested in will determine how to compose a sequence of actions to achieve a desired state transition, namely to reach a goal from a given state.

In L, we think of a plan as being a sequence of actions, and we want to determine a plan that gets us from a specified initial state to a specified final state. The basic objects of L are:

1. A set of states: $S = \{s_0, s_1, \ldots, s_n\}$.
2. A set of actions: $A = \{a_0, a_1, \ldots, a_m\}$.
3. A set of pairs of initial state and goal state. We term such a pair a *nisus*[1] and denote a set of nisi: $N =$

[1] Nisus: a striving towards a goal.

AAMAS'05, July 25-29,2005, Utrecht, Netherlands.

$\{s_0 \hookrightarrow s'_0, s_1 \hookrightarrow s'_1, \ldots, s_t \hookrightarrow s'_t\}$ where $s_i \in S$; as an abbreviation, $N = \{n_0, n_1, ..., n_t\}$ where $n_i = s_i \hookrightarrow s'_i$. $\hookrightarrow$ denotes state transition.

4. A set of plans: $P = \{p_0, p_1, .., p_s\}$. A plan p is a sequence of actions $p = a_1 \ldots a_t$ where $a_i \in A$.

Sentences in L describe the world. In particular we are interested in what actions and plans achieve.

1. The effect of an action is (a) to cause a state transition or (b) to achieve a nisus. $a \rightsquigarrow (s \hookrightarrow s')$ where $s, s' \in S$, $a \in A$ indicates action a causes a transition from s to s'. $a \rightsquigarrow n$ means action a achieves nisus n.

   In other words, actions bring about simple state transitions, and some of these state transitions may be distinguished as nisi — transitions between states we identify as start and end points for agents.

2. The effect of a plan can also be thought of in terms of state transitions: $p \rightsquigarrow (s \hookrightarrow s')$ where $p \in P$ means plan p causes a state transition from s to s'. Here p must satisfy the conditions that (a) $p = a_1 \ldots a_t$ and (b) there exists a sequence of states $s_0 s_1 \ldots s_{t-1} s_t$ such that $s = s_0$ and $s' = s_t$, $a_i \rightsquigarrow (s_{i-1} \hookrightarrow s_i)$ for $(i = 1...t)$.

   Plans are thus specific sequences of actions that, when executed in the right order, will create a path through state space between two specified states.

Our notion of plan is thus very much like the usual notion of a plan in simple AI planning. However, we choose to specify goals not by the usual target state but rather as a pair of initial state and final state (though we also use the conventional notion of goal in places). Why do we do this?

The answer is that the notion of a nisus fits rather better with our approach to deliberation than the usual notion of a goal. Obviously, plans, goals and nisi are all suitable for describing a state-transition graph $G = (S, A)$ where nodes S are states, edges A are assigned by atomic actions, and edges are directed[2]. Planning is essentially a process of finding a path between two given states s_0 and s_g, and were we to plan in classic means-ends style from initial state to goal state, or vice-versa, the usual notion of goal would suffice. However, we don't. The planning process we describe allows agents to work from start, end, or middle, and in such a situation it is convenient to be able to link start and end point exactly as a nisus does, in order to keep track of where one is in the process.

Our slightly non-standard notation for action, $a \rightsquigarrow s_1 \hookrightarrow s_2$, is similarly motivated by the deliberation process. A set of action descriptions of the kind we use can be viewed as the definition of a function $a : S \rightarrow S$. By taking an action to be a function on states in this way, we capture explicitly, and in a propositional form, the fact that the same action applied to different initial states will lead to different final states (and it is a short step to capturing non-deterministic actions).

In the rest of the paper we blur the distinction between actions and plans because we take an action a to be an atomic plan. They share the same effect of creating state transitions.

[2] An action can assign more than one edge between nodes.

3. DELIBERATION AND PLANNING

We start by considering the deliberation process that a single agent goes through. The usage that Walton and Krabbe [25] make of the term "deliberation", which is to denote the whole scope of practical reasoning, differs from that made by Bratman [2], who uses it to denote the process of choosing goals[3] that are then subject to means-ends reasoning. Since we will be considering both types of deliberation, we will denote the first by D_{WK}, and the second by D_B. For us the D_{WK} process starts with an overall nisus n_0, and uses D_B to refine the set of sub-nisi in conjunction with a process of means-ends planning.

In more detail, we recursively divide D_B and the associated planning into *phases* until a plan for n_0 is reached. All the phases share the same top-level nisus n_0, and each phase has a *deliberation context*. A deliberation context consists of 1) the top-level nisus n_0, 2) a set of intermediate nisi N_{inter} and 3) a set of useful plans P_{useful} (we will describe the way this set is constructed in detail below). Within a context, an agent deliberates in the D_B sense. Based on the result, the agent then plans. Based on the D_B and planning, the agent then decides whether or not to recursively call a *child* phase to solve a sub-problem. If this is the case, the next round of D_B and planning is delayed until the child phase is complete.

To make the procedure precise, we define the following:

1. *Justified*(n) means that a nisus n is achievable, namely it is a nisus with a plan p such that $p \rightsquigarrow n$.

2. *Src*$(N) = \{n | n \hookrightarrow n' \in N\}$ is the set of source states of a given set of nisi N.

3. *Dest*$(N) = \{s' | s \hookrightarrow s' \in N\}$ is the set of destination states of a given set of nisi N.

4. *Src*$(P) = \{s | p \rightsquigarrow (s \hookrightarrow s'), p \in P\}$ is the set of source states of a given set of plans P.

5. *Dest*$(P) = \{s' | p \rightsquigarrow (s \hookrightarrow s'), p \in P\}$ is the set of destination states of a given set of plans P.

Now, we present our first D_{WK} procedure, SD (for simple deliberation). This is called with a top-level nisus $n_0 = s_0 \hookrightarrow s_g$, initializes its top-level context with context ID $i = 0$ as $N^i_{inter} = \phi$ and a set of partial plans that might be adopted $P^i_{useful} = A$. (A is the set of actions.) SD then executes the following steps:

1. Check whether *Justified*(n_0) holds, that is whether n_o can be achieved using plans in P^i_{useful}. If it is, then stop with a plan for n_0.

2. Carry out D_B:

   (a) Create a child context ID j for its child phase.

   (b) Choose a set of intermediate nisi N^j_{inter} for the child phase from:

   Nisi of the form $s \hookrightarrow s'$ where $s \in$ *Dest*(P^i_{useful}) and $s' \in$ *Src*(P^i_{useful}). They are all the possible state pairs which connect the end state of the existing plans to the start state of the other existing

[3] To be precise Bratman considers intentions not goals, but for our purposes there is little practical difference.

plans. If there are plans for these nisi output by the planning procedure in the future, then we will have new plans by combining two existing plans with these future plans.

Nisi of the form $s_0 \hookrightarrow s'$ where $s' \in Src(P^i_{useful})$. They are all the possible state pairs which connect initial state of the top-level nisus to the start state of the existent plans.

Nisi of the form $s \hookrightarrow s_g$ where $s \in Dest(P^i_{useful})$. They are all the possible state pairs which connect the end state of the existent plans to the goal state of the top-level nisus.

Based on the candidate nisi given above, heuristics (see below) are used to gather a subset of these nisi to be N^j_{inter} which is believed to be important to achieve our top-level nisus n_0.

Note that the last two sets of nisi are not necessary, but make the process more efficient.

3. Combine plans guided by the result of D_B. For each intermediate nisus $s \hookrightarrow s' \in N^j_{inter}$, combine plans as follows
   (a) Extend forward plans that end with initial states of nisi in N^j_{inter}: look for plans $p_1 \rightsquigarrow (s \hookrightarrow s_i)$ and $p_2 \rightsquigarrow (s_i \hookrightarrow s_j)$ and combine them to give $p_1p_2 \rightsquigarrow (s \hookrightarrow s_j)$ if such p_1, s_i, s_j, p_2 exist.
   (b) Extend backward plans that start with final states of nisi in N^j_{inter}: look for plans $p_1 \rightsquigarrow (s_i \hookrightarrow s_j)$ and $p_2 \rightsquigarrow (s_j \hookrightarrow s')$ and combine them to give $p_1p_2 \rightsquigarrow (s_i \hookrightarrow s')$ if such p_1, s_i, s_j, p_2 exist.

   Add these plans into P^i_{useful}.
4. Reason about plans. Pick a subset of P^i_{useful} to be the set P^j_{useful} passed to a child phase which recursively applies the SD procedure with the new context with ID j. Repeat until all the plans in P^i_{useful} have been distributed.
5. Collect plans from child phases and combine all the P^j_{useful} into P^i_{useful}.
6. Go to the beginning of the procedure. Some termination conditions can be applied here to stop the procedure when the system figures out there is no plan to achieve n_0.

Step 2(b) is the key step in the planning process — each of the sub-parts of this step are ways in which the plan is constructed. The second and third sub-parts, respectively, capture the notions of backward chaining from the goal state and forward chaining from the initial state. The first step captures the idea that planning can work forwards and backwards simultaneously from some state in the middle of a possible plan.

Although we are describing this process for a single agent at the moment, consider how such a process might take place were several agents to be involved. In such a case different agents would be throwing out different suggestions simultaneously, and at any one time, we might have plans for achieving many different nisi "on the table". The heuristics in the fourth sub-part of 2(b), are methods that select the most promising of such a set of nisi (which can equally well be identified by a single agent) for further consideration.

Below we will adapt this procedure first to incorporate argumentation, and then to allow it to be distributed across a pair of agents. Before we do this, we obtain soundness and completeness results:

Proposition 1 (Soundness). *If SD generates a plan p, then p is a plan to achieve the top-level nisus n_0 using the atomic actions A.*

Proof. *Step 3 of the deliberation procedure ensures that only valid plans in L are composed from actions in A. Step 1 guarantees that the deliberation procedure succeeds only if there is a plan $p \rightsquigarrow n_0$. Therefore p is a valid plan to achieve n_0 using the atomic actions A.* □

Before attempting the completeness result, we need some further notation:

1. P_n is the set of plans with n actions in its action sequence.
2. $\oplus$ is a plan combination operator $P_i \oplus P_j = \{p_1p_2 | p_1 \in P_i \wedge p_2 \in P_j\} \cup \{p_2p_1 | p_1 \in P_i \wedge p_2 \in P_j\}$ where p_1p_2 and p_2p_1 must satisfy the valid plan conditions given above. This operator corresponds to step 3.

Proposition 2 (Completeness). *If there is a plan for intial nisus n_0, then SD will succeed with a plan p which achieves n_0.*

Proof. *In step 2, P^i_{useful} determines N^j_{inter} (j is a child context of i). In step 3, N^j_{inter} determines the plans being added into P^i_{useful}. Therefore step 2 and 3 together determine the growth of P^i_{useful}. The recursive child phase calls in step 5 expedite the discovery of the top-level nisus n_0; they don't affect the growth of P^i_{useful}. We will show that step 2 and step 3 together will grow P^0_{useful} to contain all the plans which can be generated from atomic actions A so that if there is a plan for n_0 the deliberation procedure will certainly discover it.*

We divide P^0_{useful} into n disjoint subsets $P = P_1 \cup P_2 \cup ... \cup P_n$, where n is the number of states(namely length of plans in P_n is in the range of $1 \ldots n$)[4]. Initially $P^0_1 = A$ and $P^0_i = \phi$ for $i = 2 \ldots n$, each time step 2 and step 3 together grow P in the following way: $P^{k+1}_t = \cup(P^k_i \oplus P^k_j)$ for all $i + j = t$. Therefore if, in step 2 and 3, P^k_i are fixed for $i = 1 \ldots t-1$ then $P^{k+1}_t = P^k_t$. P_1 is fixed during any iteration; after the first iteration P_2 is fixed since P_1 is fixed; after the second iteration P_3 is fixed since P_1, P_2 are fixed. In this way, after $n-1$ iterations, P_n will be fixed. Since the maximal plan length is n, P will contain all the possible plans after $n-1$ iterations. Therefore if there is a valid plan for n_0 then P will contain it after $n-1$ iterations. This ends the proof. □

[4]In any plan, we discard any action sequence that includes a cycle $p = a_1a_2 \ldots a_{i-1}a_i \ldots a_ia_{i+1} \ldots$, because the corresponding contracted plan $p' = a_1a_2 \ldots a_{i-1}a_ia_{i+1} \ldots$ is guaranteed to be reached sooner or later by the procedure where we have two plans $p_1 = p' = a_1a_2 \ldots a_{i-1}$ and $p_2 = a_ia_{i+1} \ldots$ Doing this effectively makes the Markov assumption, taking the effects of an action to uniquely determine the succeeding state.

4. ARGUMENT AND DELIBERATION

To combine D_B with argumentation, we need to do three things. First, we extend L with predicates that control the D_B procedure. Second, we establish logic-based rules for nisus reduction, reasoning about plans, plan combination and information passing through different contexts. (This will enable us to construct plans by STRIPS-like logical reasoning). Third, we add a commitment store [11] to track the course of D_B and planning.

With a knowledge base expanded using the extended L, a plan for a nisus is certainly contained in the theorems of a subset of the knowledge base. However, the deliberation problem, to some extent, is to select an efficient way to construct a proof which backs up a plan for a nisus (the proof then becomes the justification that can be provided in a multi-agent D_{WK}). The commitment store provides a trace of how such a proof is constructed.

4.1 Additional notation

To capture the context of phases, we introduce the following predicates into L

1. $\mathit{Ultimate}(n)$ denotes $n \in N$ is the top-level nisus.

2. $N(\mathit{contextid}, n)$ denotes $n \in N$ is an intermediate nisus in context with ID $\mathit{contextid}$. It is a predicate that determines whether $n \in n_{inter}^{contextid}$.

3. $P(\mathit{contextid}, p)$ denotes $p \in P$ is a useful plan in context with ID $\mathit{contextid}$. It is a predicate that determines whether $p \in P_{useful}^{contextid}$ or not.

4. $\mathit{Justified}(\mathit{contextid}, n)$ denotes the existence of a plan for nisus n in a the context $\mathit{contextid}$.

5. $\mathit{Parent}(id_1, id_2)$ denotes the fact that context id_1 is the parent of context id_2.

4.2 Rules

In order to create arguments that support plans, we need to be able to trace the planning process. To do that we need to introduce the following logical rules.

Nisus justification

$$P(i,p) \wedge [p \rightsquigarrow (s' \hookrightarrow s')] \rightarrow \mathit{Justified}(\mathit{contextid}, s \hookrightarrow s')$$

Note that here, as in all these rules, $\rightarrow$ denotes material implication.

Candidate nisus composition

$$\mathit{Ultimate}(n) \quad \rightarrow \quad N(j,n)$$

$$\begin{array}{l} \mathit{Parent}(i,j) \\ \wedge P(i,p_1) \\ \wedge P(i,p_2) \\ \wedge [p_1 \rightsquigarrow (s' \hookrightarrow s_{foo1})] \\ \wedge [p_2 \rightsquigarrow (s_{foo2} \hookrightarrow s)] \quad \rightarrow \quad N_{cand}(j, s \hookrightarrow s') \end{array}$$

$$\begin{array}{l} \mathit{Parent}(i,j) \\ \wedge P(i,p) \\ \wedge \mathit{Ultimate}(s_0 \hookrightarrow s_g) \\ \wedge [p \rightsquigarrow (s \hookrightarrow s_{foo})] \quad \rightarrow \quad N_{cand}(j, s_0 \hookrightarrow s) \end{array}$$

$$\begin{array}{l} \mathit{Parent}(i,j) \\ \wedge P(i,p) \\ \wedge \mathit{Ultimate}(s_0 \hookrightarrow s_g) \\ \wedge [p \rightsquigarrow (s_{foo} \hookrightarrow s)] \quad \rightarrow \quad N_{cand}(j, s \hookrightarrow s_g) \end{array}$$

We can use heuristics to select $N(j,n)$ from $N_{cand}(j,n)$ in order to reduce the search space. Without the heuristics, we will have rule

$$N_{cand}(j,n) \rightarrow N(j,n)$$

so that every candidate nisus is considered.

Candidate plan combination

$$\begin{array}{l} [p_1 \rightsquigarrow (s \hookrightarrow s_m)] \\ \wedge [p_2 \rightsquigarrow (s_m \hookrightarrow s')] \quad \rightarrow \quad p_1 p_2 \rightsquigarrow (s \hookrightarrow s') \end{array}$$

$$\begin{array}{l} P(i,p_1) \\ \wedge P(i,p_2) \\ \wedge N(i, s \hookrightarrow s_{foo}) \\ \wedge p_1 p_2 \rightsquigarrow (s \hookrightarrow s') \quad \rightarrow \quad P(i, p_1 p_2) \end{array}$$

$$\begin{array}{l} P(i,p_1) \\ \wedge P(i,p_2) \\ \wedge N(i, s_{foo} \hookrightarrow s) \\ \wedge p_1 p_2 \rightsquigarrow (s \hookrightarrow s') \quad \rightarrow \quad P(i, p_1 p_2) \end{array}$$

Plan selection

$$\mathit{Parent}(i,j) \wedge P(i,p) \quad \rightarrow \quad P_{cand}(j,p)$$

Again we can use heuristics select $P(j,p)$ from $P_{cand}(j,p)$. Without using heuristics, we will have rule

$$P_{cand}(j,p) \rightarrow P(j,p)$$

so that every candidate plan is considered.

Plan collection

$$P(j,p) \wedge \mathit{Parent}(i,j) \quad \rightarrow \quad P(i,p)$$

These basic rules provide a backbone to guarantee that our procedure searches the whole space of plans so that if there is a plan to achieve the nisus n_0 then we will reach it sooner or later.

4.3 Heuristics

The basic rules give us a no-frills planning procedure. Adding in heuristics like those given here tries to ensure that if there is a plan that can achieve the top-level nisus, the deliberation procedure will reach it as early as possible. We take inspiration from decision-theoretic planning [1], where action choices are made on the basis of their expected cost. Accordingly we introduce the following notions of cost.

1. The *action-state transition cost* $\mathit{cost}(a, s, s')$ is the cost of taking action a to transform state from s to s'. The value is computed or assigned outside the resoning system.

2. The *plan-state-transition cost* $\mathit{cost}(p, s, s')$ is the cost of taking a plan p to transform state from s to s'. The value is computed from $\mathit{cost}(a, s_i, s'_{i+1})$ for all actions a in the plan p.

3. The *overall cost* $cost(s \hookrightarrow s')$ is an abbreviation of $cost(s, s')$, the cost of transforming state s into s'. The value is computed from $cost(p, s, s')$ for all the plans p and actions a which can cause state transition from s to s'.

The idea is that although we often want to consider $p \rightsquigarrow (s \hookrightarrow s')$ as a holistic entity, to do D_B and planning we need to make comparisons between plans and actions, and we use costs to make these comparisons.

The cost of a plan can be derived from the cost of its actions in the same kind of way as it is done in decision-theoretic planning[5]. Whatever mechanism is adopted, it is outside the logical reasoning that we are studying here. Thus the assignment of costs to overall plans is, so far as the D_B and planning processes are concerned, carried out by an oracle.

Another useful notion in deciding which nisi to adopt is the *correlated valuation* of one nisus relative to another, denoted $value(n, n')$. This captures the value of achieving nisus n' in order to achieve n, and can be computed from the costs of all the plans which have the form $p \rightsquigarrow n$ for which there exists a subplan p' of p such that $p' \rightsquigarrow n'$. If there is no such a sub plan p' then $value(n, n') = 0$.

With these ideas in place, we can suggest heuristics for plan selection and nisus composition. One possibility for plan selection is to select the lowest cost plan:

$$\begin{aligned}&Parent(i, j)\\ &\wedge P_{cand}(j, p)\\ &\wedge P(i, p)\\ &\wedge P(i, p')\\ &\wedge cost(i, p) < cost(i, p') \quad \rightarrow \quad P(j, p)\end{aligned}$$

A possibility for nisus composition is to only adopt nisi for which the correlated valuation is above some threshold. To do this we can use:

$$\begin{aligned}&N(j, n)\\ &\wedge N_{cand}(j, n')\\ &\wedge value(g, g') > c \quad \rightarrow \quad G(j, g')\end{aligned}$$

Other heuristics can, of course, be adopted.

4.4 Single agent deliberation

We are now in a position to explain how a single agent can use argumentation-based D_{WK} to figure out what to do. We assume the agent has a knowledge base KB which contains a description of the physical world (e.g. the set of available actions and their effects). The agent also has a commiment store CS which it uses to trace the course of deliberation. The idea behind the procedure is to guarantee that all neccessary sentences to support an argument are available in the commiment store CS before such an argument is constructed. The argumentation system used is

$$\mathcal{AS} = \langle \mathcal{A}(KB \cup CS, Undercut, Pref \rangle$$

in the notation of [19].

The procedure for argumentation-based deliberation, SDA (Simple Deliberation through Argumentation), is given a top-level nisus $n_0 = s_0 \hookrightarrow s_g$. It first initializes the context id with $i = 0$, CS with $Ultimate(n_0)$ and $P(i, a)$ for all $a \in A$.

[5]Such costs may be assigned by a form of reinforcement learning, for example.

1. Check $\mathcal{AS}$ to see whether $Justified(n_0)$ is acceptable. If it is, then stop with a plan for n_0 in CS.

2. Carry out a D_B:

   (a) Set a context ID j for a child phase.

   (b) Using KB and CS, use $\mathcal{AS}$ to check if $N(j, n)$ is acceptable for nisi of the following three cases:

   Nisi such as $(s \hookrightarrow s')$ for $s \in Dest(P)$ and $s' \in Src(P)$. This captures the idea of extending existent plans forwards and backwards.

   Nisi such as $(s_0 \hookrightarrow s')$ for $s' \in Src(P)$. This captures the idea of extending the existent plans forwards from the source of the top-level nisus.

   Nisi such as $(s \hookrightarrow s_g)$ for $s \in Dest(P)$ to capture the idea of extending the existent plans backwards from the destination of the top-level nisus.

   Assert all the acceptable sentences $N(j, n)$ in CS. Notice that the above can be achieved only if the rules for nisus composition are used. If the agent exhausts all the possible cadidates nisi $N_{cand}(j, n)$ but no $G(j, n)$ can be asserted, then the child phase returns to the parent context with no new plans added.

3. Combine plans guided by the results of D_B. For all the nisi $n = (s \hookrightarrow s')$ with acceptable arguments for $N(j, n)$, combine plans as follows:

   Plans from s. Look for plans $p_1 \rightsquigarrow (s \hookrightarrow s_i)$ and $p_2 \rightsquigarrow (s_i \hookrightarrow s_j)$, combine them to be $p_1p_2 \rightsquigarrow (s \hookrightarrow s_j)$ if such p_1, s_i, s_j, p_2 exist.

   Plans to s'. Look for plans $p_1 \rightsquigarrow (s_i \hookrightarrow s_j)$ and $p_2 \rightsquigarrow (s_j \hookrightarrow s')$, combine them to be $p_1p_2 \rightsquigarrow (s_i \hookrightarrow s')$ if such p_1, s_i, s_j, p_2 exist.

   Check with $\mathcal{AS}$, assert all acceptable $p_1p_2 \rightsquigarrow (s \hookrightarrow s_j)$, $p_1p_2 \rightsquigarrow (s_i \hookrightarrow s')$ into CS.

4. Reason about plans. Using the plan selection rules, identify the candidate useful plans p, and for each use $\mathcal{AS}$ to check whether $P(j, p)$ is acceptable. If it is acceptable, assert it into CS.

5. Recursively call a child phase to go through SDA with the new context with ID j.

6. Collect plans from the child phase: Use plan collection rules to identify candidates plans p. For each p, check with $\mathcal{AS}$ whether $P(i, p)$ is acceptable. If it is acceptable, assert it into CS.

7. Go to the beginning of the procedure.

Since SDA is based on a process for D_B and planning which we know is sound and complete, we can easily show that it is sound and complete itself:

PROPOSITION 3 (SOUNDNESS). *If plan p for nisus n_0 is acceptable according to AS at the end of the SDA, then p achieves n_0 using actions from A.*

PROOF. *Step 3 combines plan p from P_1 and P_2 only if AS accepts the combination. Thus AS accepts the effects of p. Step 1 ensures that p is a plan for n_0.* □

PROPOSITION 4 (COMPLETENESS). *If there is a plan p which achieves nisus n_0 using atomic actions A, then SDA will generate p.*

PROOF. *Similar to the proof of Proposition 2, since steps 2 and 3 together will explore all the possible combinations of discovered acceptable plans, the procedure will generate all the plans acceptable by $\mathcal{AS}$ at the end of the procedure. If there is plan p which achieves initial nisus n_0 using actions from A, then such a plan is contained in all the acceptable plans by $\mathcal{AS}$.* □

5. DELIBERATION DIALOGUES

We now consider how to extend the D_{WK} process to become a dialogue. We describe the dialogue process as being between just two agents, but it can easily be extended to a multi-party dialogue.

5.1 Basic configuration

The scenario for which our deliberation dialogue was created is as follows:

1. Dialogues take place between two agents, $A1$ and $A2$.
2. $A1$ has initial knowledge base KB_1 and commitment store CS_1.
3. $A2$ has initial knowledge base: KB_2 and a commiment store CS_2.
4. $A1$ and $A2$ share the same rules for planning and deliberation and differ only in the way they evaluate plans and nisi. They may, however, have different set of actions reflecting different capabilities.
5. The context ID, id, is shared by $A1$ and $A2$. Initially, $id = 0$.
6. $A1$ and $A2$ have an mechanism to allocate unique context IDs.
7. Both $A1$ and $A2$ can access CS_1 and CS_2, hence the argumentation system of $A1$ is

   $$\mathcal{AS}_1 = \langle \mathcal{A}(KB_1 \cup CS_1 \cup CS_2), Undercut, Pref \rangle$$

   and the argumentation system of $A2$ is

   $$\mathcal{AS}_2 = \langle \mathcal{A}(KB_2 \cup CS_1 \cup CS_2), Undercut, Pref \rangle.$$

Within this scenario we need to add the following idea of an auxilliary sub-dialogue.

5.2 Auxiliary discussion sub-dialogue

One of the reasons for agents to enagage in deliberation dialogues is to combine both agents' reasoning and planning capabilities. One possible downside, the fact that conflicts may arise between the two agents, can be resolved by the use of argumentation (which, at heart, is a system for resolving conflicts in terms of the acceptability of the arguments which support the conflicting statements in two argumentation systems). To achieve this resolution we need an auxiliary discussion sub-dialogue to render a sentence acceptable to both agents, by which we mean that it is accepted by the argumentation systems $\mathcal{AS}_1$ and $\mathcal{AS}_2$.

A discussion sub-dialogue is started by a dialogue move $discuss(p)$. Assume that $A1$ moves first, the discussion sub-dialogue proceeds as follows:

1. $A1$ checks with its own argumentation system AS_1 whether p is acceptable, if it is, then $A1$ makes the locution $discuss(p)$, indicating that p is open for discussion.
2. $A2$ checks with its argumentation system AS_2 whether p is acceptable, if it is, A_2 stops and declares that p is accepted by both agents. Otherwise, $A2$ challenges p indicating that it needs to see the argument for p (which will be the reason behind A_1's suggestion of p).
3. $A1$ responds to the challenge by asserting the set of support S for p.
4. For each sentence $q' \in S$, $A2$ checks with AS_2. For the unaccepted sentences $q' \in S$, $A2$ discusses $\neg q'$ with $A1$, if any of the $\neg q'$ are accepted by the discussion then go to next step; otherwise stop and declare p is accepted by both agents.
5. $A1$ replaces S with another alternative support and goes back to step 3.
6. If $A1$ exhausts all the possible supports for p but the discussion with $A2$ accepts none of them, then declare p is not accepted by the discussion.

With this machinery, we can now set down the deliberation dialogue.

5.3 A dialogue for deliberation

Two agents can have two different set of atomic actions. This means they have different capability or different views of the physicial world.

We assume that initially $A1$ and $A2$ agree on a top-level nisus $n_0 = s_0 \hookrightarrow s_g$ (though they could arrive at this after another dialogue, a negotiation perhaps). $A1$ and $A2$ initialize the context id with $i = 0$. $A1$ initializes CS_1 with $Ultimate(n_0)$ and $P(i, a)$ for all its actions, and $A2$ does the same for its commitment store. The simple deliberation dialogue, SDD, then consists of the following steps:

1. $A1$ discusses $A2$ to check whether $Justified(n_0)$ is acceptable. If it is, then stop with a plan for n_0.
2. Carry out D_B:
   (a) Create a context ID j for a child phase.
   (b) $A1$ discusses with $A2$ to check whether $N(j, n)$ is acceptable for nisus of the following three cases

   Nisi of the form $s \hookrightarrow s'$ where $s \in Dest(P)$ and $s' \in Src(P)$, to capture the ideas of extending existent plans forwards and backwards.

   Nisi of the form $s_0 \hookrightarrow s'$ where $s' \in Src(P)$, to capture the ideas of extending the existent plans forwards from the source of the top-level nisus.

   Nisi of the form $s \hookrightarrow s_g$ where $s \in Dest(P)$, to capture the ideas of extending the existent plans backwards from the destination of the top-level nisus.

   Assert all the acceptable sentences $N(j, n)$ in CS_1. Notice that the above can be achieved only if nisus composition rules on are used.

(c) $A2$ carries out the same step as $A1$ but asserts results into CS_2. The bidirectional discussion is to ensure all the sentences that will be needed in the future can be exchanged between $A1$ and $A2$.

After $A1$ and $A2$ have exhausted all the nisi n that satisfy $N_{cand}(j, n)$, and no new $N(j, n)$ are asserted, return to the parent context with no new plans are asserted.

3. Combine plans guided by the results of D_B.

   (a) For all the nisi $n = (s \hookrightarrow s')$ with acceptable arguments for $N(j, n)$, $A1$ combines plans:

   Plans from s: look for plans $p_1 \rightsquigarrow (s \hookrightarrow s_i)$ and $p_2 \rightsquigarrow (s_i \hookrightarrow s_j)$, combine them to be $p_1p_2 \rightsquigarrow (s \hookrightarrow s_j)$ if such p_1, s_i, s_j, p_2 exist.

   Plans to s': look for plans $p_1 \rightsquigarrow (s_i \hookrightarrow s_j)$ and $p_2 \rightsquigarrow (s_j \hookrightarrow s')$, combine them to be $p_1p_2 \rightsquigarrow (s_i \hookrightarrow s')$ if such p_1, s_i, s_j, p_2 exist.

   $A1$ discusses $p_1p_2 \rightsquigarrow (s \hookrightarrow s_j)$, $p_1p_2 \rightsquigarrow (s_i \hookrightarrow s')$, with $A2$, asserting the acceptable plans into CS_1.

   (b) $A2$ does the same as $A1$ does for (a) but asserts the results into CS_2.

4. Reason about plans:

   (a) $A1$ uses the plan selection rules to identify candidate useful plans p, and discusses with $A2$ if $P(j, p)$ is acceptable. If it is acceptable, $A1$ asserts it into CS_1.

   (b) $A2$ carries out the analogous process.

5. Recursively call a child phase to go through SDD with the new context with ID j.

6. Collect plans from the child phase:

   (a) $A1$ uses the plan collection rules to figure out the candidates of plans p should be collected. Then he discusses with $A2$ to check whether $P(i, p)$ is acceptable. If it is acceptable, assert it into CS_1.

   (b) $A2$ carries out the analogous process

7. Go to the beginning of the procedure.

Once again we can prove the soundness and completeness of the procedure, showing that reacsting it as a dialogue does not detract from it:

PROPOSITION 5 (SOUNDNESS). *If plan p for nisus n_0 is acceptable by both agents at the end of SDD, then p achieves n_0 using atomic actions that both agents agree upon.*

PROOF. *Step 3 combines plan p from P_1 and P_2 only if $\mathcal{AS}_1$ and $\mathcal{AS}_2$ both accept the combination. Thus both agents agree on the effects of p. Step 1 ensures that both agents agree that p is a plan for n_0.* □

PROPOSITION 6 (COMPLETENESS). *If there is a plan p which achieves nisus n_0 using a set of atomic actions that both agents agree upon, then SDD will generate p.*

PROOF. *Similar to the proof of Propositions 2 and 4, since steps 2 and 3 together will explore all the possible combinations of discovered plans accepted by both agents, the procedure will generate all the plans acceptable by both $\mathcal{AS}_1$ and $\mathcal{AS}_2$ at the end of the procedure. If, according to the acceptable atomic actions A agreed by both agents, there is plan p which achieves initial nisus n_0, then such a plan is contained in all the acceptable plans by $\mathcal{AS}_1$ and $\mathcal{AS}_2$.* □

6. DISCUSSION

The full argumentation-based deliberation dialogue successfully composes plans in one of the following ways:

1. $A1$ composes the whole plan, $A2$ agrees with it.

2. $A2$ composes the whole plan, $A1$ agrees with it.

3. $A1$ composes some parts of the plan; $A2$ composes some other parts of the plan; $A1$ and $A2$ combine the two parts at the end, and both $A1$ and $A2$ agree with the final plan.

Thus we can see that our process combines the "create a plan and then convince others it works" approach of [17] with the "merge different plans" approach of [7]. For a given situation SDD will typically take an approach that is a mixture of the two, involving the creation and merging of separate subplans some of which one agent has to persuade the other to adopt. SDD thus achieves the nisus that we set out at the start of the paper.

One thing to note about this work concerns the heuristics used to guide the search for plans during deliberation. These are never specified in any detail, though we give some high-level hints about the possible form that they may take. This does not detract from the formal results, since the results hold even if we have no heuristics (in which case we essentially do an exhaustive search through the full state-space). However, decent heuristics will help to focus the search and thus make it more efficient.

7. CONCLUSION

This paper has described a mechanism for carrying out deliberation dialogues in the sense of Walton and Krabbe [25] — that is dialogues in which agents decide what to do. Our approach, which as described is limited to two agents but could easily be generalised, recursively mixes nisus selection and planning, allowing these tasks to be distributed between the agents in a flexible way. The approach makes it possible for agents to combine their knowledge about the environment, and to make use of the planning abilities of both agents (since one can readily imagine that they have complementary expertise, as embodied in the heuristics they can employ).

Two directions of furture research are particularly attractive to us. First, as mentioned above, it seems appropriate to allow the agents to learn the values of actions across a number of trials, and this might easily be achieved by techniques from reinforcement learning. Doing so suggests a bridge between the kind of procedure we have developed here and multi-agent decision theoretic planning of the kind considered in [10]. Exploring such connections is the second direction we intend to take.

Acknowledgements

This work was made possible by funding from NSF #REC-02-19347, NSF #IIS 0329037 and EU FP6-IST 002307 (ASPIC).

8. REFERENCES

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

[2] M. E. Bratman. *Intention, plans, and practical reason*. CSLI Publications, 1999.

[3] D. D. Corkill. Hierarchical planning in a distributed environment. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 168–175, 1979.

[4] M. E. desJardins, E. H. Durfee, C. L. Ortiz, and M. J. Wolverton. A survey of research in distributed, continual planning. *AI Magazine*, 21(1), 2000.

[5] 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.

[6] E. H. Durfee and V. R. Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, KDE-1:63–83, 1991.

[7] E. Ephrati and J. S. Rosenschein. Multi-agent planning as the process of merging distributed sub-plans. In *Proceedings of the Twelfth International Workshop on Distributed Artificial Intelligence*, pages 115–129, 1993.

[8] K. Greenwood, T. Bench-Capon, and P. McBurney. Structuring dialogue between the People and their representatives. In R. Traunmüller, editor, *Electronic Government: Proceedings of the Second International Conference (EGOV03), Prague, Czech Republic*, Lecture Notes in Computer Science 2739, pages 55–62, Berlin, Germany, 2003. Springer.

[9] B. J. Grosz and S. Kraus. The evolution of sharedplans. In M. J. Wooldridge and A. Rao, editors, *Foundations of Rational Agency*, volume 14 of *Applied Logic*. Kluwer, The Netherlands, 1999.

[10] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*, pages 1523–1530, 2001.

[11] C. L. Hamblin. *Fallacies*. Methuen and Co Ltd, London, UK, 1970.

[12] D. Hitchcock, P. McBurney, and S. Parsons. The eightfold way of deliberation dialogues. *International Journal of Intelligent Systems*, 2004.

[13] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[14] H. Levesque, P. Cohen, and J. Nunes. On acting together. In *Proceedings of the National Conference on Artificial Intelligence*, pages 94–99, 1990.

[15] N. Maudet and F. Evrard. A generic framework for dialogue game implementation. In *Proceedings of the 2nd Workshop on Formal Semantics and Pragmatics of Dialogue*, University of Twente, The Netherlands, May 1998.

[16] P. McBurney. *Rational Interaction*. PhD thesis, University of Liverpool, 2002.

[17] P. Panzarasa, N. R. Jennings, and T. J. Norman. Formalising collaborative decision making and practical reasoning in multi-agent systems. *Journal of Logic and Computation*, 12(1):55–117, 2002.

[18] S. Parsons and P. McBurney. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation*, 12(5):415–439, 2003.

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

[20] C. Reed. Dialogue frames in agent communications. In Y. Demazeau, editor, *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 246–253. IEEE Press, 1998.

[21] M. Schroeder, D. A. Plewe, and A. Raab. Ultima ratio: should Hamlet kill Claudius. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 467–468, 1998.

[22] K. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the Eleventh Joint Conference on Artificial Intelligence*, pages 517–523, 1989.

[23] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[24] W. van der Hoek and M. Wooldridge. Tractable multiagent planning for epsitemic goals. In *Proceedings of the 1st International Conference on Autonomous Agents and Multiagent Systems*, 2002.

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

[26] R. Zlot and A. Stentz. Market-based multirobot coordination using task abstraction. In *Proceedings of the 4th International Conference on Field and Service Robotics*, 2003.