Agreeing on Plans Through Iterated Disputes

Alexandros Belesiotis
School of Informatics
The University of Edinburgh
Edinburgh EH8 9AB, UK
A.Belesiotis@ed.ac.uk

Michael Rovatsos
School of Informatics
The University of Edinburgh
Edinburgh EH8 9AB, UK
Michael.Rovatsos@ed.ac.uk

Iyad Rahwan*
[1]Masdar Institute of Science & Technology, Abu Dhabi, UAE
[2](Visiting Scholar) MIT, USA
[3](Fellow) The University of Edinburgh, UK

ABSTRACT

Autonomous agents transcend their individual capabilities by cooperating towards achieving shared goals. The different viewpoints agents have on the environment cause disagreements about the anticipated effects of plans. Reaching agreement requires the resolution of such inconsistencies and the alignment of the agents' viewpoints.

We present a dialogue protocol that enables agents to discuss candidate plans and reach agreements. The dialogue is based on an argumentation process in the language of situation calculus. Agreement is reached through persuasion, thereby aligning the planning beliefs of the agents.

We describe our abstract iterated dialogue protocol, and extend it for the specific problem of arguing about plans. We show that our method always terminates and produces sound results. Furthermore, we detail a set of extensions to simplify reasoning and reduce the exchanged information.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Theory

Keywords

Argumentation, Planning, Communication protocols, Distributed problem solving

1. INTRODUCTION

In many collaborative practical decision-making settings it is necessary for cooperating agents to come up with proposals for joint action in order to achieve goals that transcend their individual capabilities. Reaching agreement on plans for action may be hindered if different agents anticipate a plan to affect the environment in different ways. Such disagreements can be the result of the locality of sensing, outdated information, contradicting domain beliefs encoded by different agent designers, or simply because different agents have conducted different inferences and therefore their beliefs are not aligned. Since cautious autonomous agents will not subscribe to plans that they consider to be harmful, ineffective or questionable, methods for reaching agreement in a collaborative planning environment are needed, and these will involve a mechanism that enables the agents to discuss and identify plans that the agents will follow.

In this paper we deal with the problem of identifying and agreeing on a plan that satisfies all agents. We extend the dialogue protocol presented in [2], providing a framework that allows cooperative agents to search the space of potential plan proposals, resolve contradictory beliefs and reach agreement while aligning their knowledge.

Our system is based on the combination of argumentation theory and the language of situation calculus in a distributed setting. The dialogue-based approach enables the distribution of the argumentation process, and allows all agents to initiate discussion about the candidate plans they have generated. The resolution of contradictions is restricted to beliefs that are relevant to concrete plans. Avoiding the merging of all beliefs and the resolution of all inconsistencies is particularly significant for agents with extensive domain knowledge or privacy concerns.

We present an abstract argument-based protocol that enables discussion of candidate proposals and extend it for the specific problem of arguing about plans. The dialogue is broken down into sub-dialogues, which discuss alternative proposals. If a sub-dialogue fails, the protocol ensures that the source of the disagreement is discovered and resolved, and that the knowledge of the agents is gradually aligned as participants' local misconceptions are uncovered. We provide termination and soundness results for the suggested protocol and discuss extensions that may support the early identification of mutually acceptable plan proposals.

The remainder of this paper is structured as follows: Section 2 introduces our abstract dialogue protocol and discusses its important properties. Section 3 extends the protocol for the specific problem of arguing about plans. Section 4 details extensions that can be employed to guide reasoning or reduce the size of the arguments about plans. Section 5 overviews related work and concludes.

2. ARGUMENTATION MODEL

Argumentation theory provides strong theoretical foundations for formally defining the notion of acceptability, and mechanisms for the identification and resolution of contradictions in the agents' knowledge. We start by describing

*This work commenced when Iyad Rahwan was with the British University in Dubai

Cite as: Agreeing on Plans Through Iterated Disputes, Alexandros Belesiotis, Michael Rovatsos and Iyad Rahwan, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

our two-agent[1] dialogue framework at the level of abstract argumentation [3], together with a protocol for iterated argumentation that is suitable for arguing over plan proposals as we will later show.

2.1 Abstract Argumentation

Our basic argumentation framework follows [3]:

DEFINITION 1. *An* argumentation framework *is a pair* $AF = \langle \mathscr{A}, \mathscr{R} \rangle$ *where* $\mathscr{A}$ *is a set of arguments and* $\mathscr{R} \subseteq \mathscr{A} \times \mathscr{A}$ *is the binary defeats relation between arguments. We say that* $A \rightarrow B$ *iff* $(A, B) \in \mathscr{R}$.

For $AF = \langle \mathscr{A}, \mathscr{R} \rangle$, a set $S \subseteq \mathscr{A}$ is said to be conflict-free if there are no arguments $A, B \in S$ *s.t.* $B \rightarrow A$. We will consider a preference ordering over $\mathscr{A}$. For $A, B \in \mathscr{A}$, $pLevel(A) > pLevel(B)$ will denote that A has a higher preference than B. The preference level resolves ties between arguments with contradictory claims. Let $AF = \langle \mathscr{A}, \mathscr{R} \rangle$ be an argumentation framework. An argument $A \in \mathscr{A}$ is *acceptable with respect to a set of arguments* $\mathcal{S}$ iff for all arguments B in $\mathscr{A}$, where $B \rightarrow A$, there exists an argument $C \in \mathcal{S}$ *s.t.* $C \rightarrow B$. Grounded (sceptical) semantics are defined using the characteristic function.

DEFINITION 2. *The* characteristic function *of an argumentation framework* $AF = \langle \mathscr{A}, \mathscr{R} \rangle$*, is the function* $\mathscr{F}_{AF} : 2^{\mathscr{A}} \rightarrow 2^{\mathscr{A}}$*, which is defined as follows:* $\mathscr{F}_{AF}(\mathcal{S}) = \{A \mid A \textit{ is acceptable } w.r.t.\ \mathcal{S}\}$.

DEFINITION 3. *The grounded extension of an argumentation framework* AF*, denoted by* GE_{AF} *is the least fixed point of* $\mathscr{F}_{AF}$.

Consider the sequence: $\mathscr{F}^0_{AF} = \emptyset$, $\mathscr{F}^{i+1}_{AF} = \{A \in \mathscr{A} \mid A \text{ is acceptable w.r.t. } \mathscr{F}^i_{AF}\}$. According to [3], it holds that: all arguments in $\cup_{i=0}^{\infty}(\mathscr{F}^i_{AF})$ are in GE_{AF}, and if each argument is defeated by at most a finite number of arguments, then an argument is in GE_{AF} iff it is in $\cup_{i=0}^{\infty}(\mathscr{F}^i_{AF})$.

We employ grounded semantics in order to shift the burden of proof for plans to the agent that proposed them and assert that a plan will be accepted iff its proponent can defend it against all possible attacks.

2.2 Dialogue Protocol

In this section we describe the *iterated disputes* protocol, which involves sequences of *disputes* [4], with each dispute discussing an alternative proposal. Every iteration is followed by an argument revision step which aligns the argument sets of the agents. We consider the agents to have distinct argument sets, instead of sharing the same pool of arguments, which is usually the case in disputes.

2.2.1 Iterated Disputes

The agents evaluate the acceptability of a proposal through a *dispute* [4]. The agent that made the proposal plays the role of the proponent PRO, leaving the role of opponent OPP to the other party. The proponent is responsible for constructing arguments in favour of the proposal, while the opponent's role is to show that the proposal should not be accepted. The game progresses with agents presenting arguments defeating the arguments of their rival.

Iterated disputes facilitate the discussion of different proposals in sequence. This protocol extends argument games enabling discussion about different proposals in a dialogue.

Let $AF = \langle \mathscr{A}, \mathscr{R} \rangle$ be an argumentation system, with $\mathscr{A} = \mathscr{A}_{PRO} \cup \mathscr{A}_{OPP}$ the union of the arguments that are available to the proponent and the opponent respectively. A *dispute tree for some argument* P *in* $\mathscr{A}$, denoted by $\mathcal{T}$, is a tree with root P whose vertices and edges are subsets of $\mathscr{A}$ and $\mathscr{R}$, respectively. The edges in a dispute tree are directed from vertices to their parent node. $Depth(\mathcal{T}, A)$ for an argument A in a dispute tree $\mathcal{T}$ denotes the number of edges in the dispute line from A to the root of the tree. $Children(\mathcal{T}, V, \delta)$ represents all arguments in $\mathcal{T}$ that have V, which is located in depth δ, as their parent node.

A *dispute line* is a path in the dispute tree: $T = V_k \rightarrow \ldots \rightarrow V_1 \rightarrow V_0 = P$. A dispute line is called *open*/*closed* if the agent who has to make the following move is able/unable to attack the other party's most recent move. A closed dispute line for some argument P is a *failing defence/attack of* P if the leaf node argument move has been made by the opponent/proponent.

A *dispute* for some argument P is a sequence of *moves*: $d = \langle \mu_1, \mu_2, \ldots, \mu_k, \ldots \rangle$ affecting a dispute tree that has P as its root. Every dispute evaluates the acceptability of a candidate proposal. Dispute d after k moves will be denoted as d_k. The state of the dispute d_k is a tuple $State(d_k) = \langle \mathcal{T}_k, V_k, CS^{PRO}_k, L_k, Q_k, \mathscr{A}_k \rangle$. $\mathcal{T}_k$ is the dispute tree after the most recent argument move V_k. CS^{PRO}_k contains arguments that have been presented by the proponent in the current dispute line providing defence on P. L_k is the set of arguments the proponent has presented in the current dispute tree. Q_k contains sets of arguments presented by the proponent that failed to defend P. $\mathscr{A}_k$ represents the arguments that have been exchanged in the dispute by both agents, excluding the proposal argument. We will refer to a dispute as being *closed* if the agent who has to make the following move is unable to make a move affecting the tree of the dispute. A closed dispute line for some argument P is a *failing defence/attack* if the final move affecting the dispute tree was made by the opponent/proponent.

An *iterated dispute* is a sequence of disputes $\mathscr{D} = \langle d_1, d_2, \ldots, d_l, \ldots \rangle$. An iterated dispute can be rewrtitten as a sequence of *legal* moves $\mathscr{D} = \langle \mu_{1,0}, \mu_{1,1}, \ldots, \mu_{l,k}, \ldots \rangle$, where $\mu_{l,k}$ denotes the kth move of the lth dispute. The boolean function $Legal(\mu, \mathscr{D})$ succeeds if all the *conditions* specified by the move hold before the move is applied. $CurrentDispute(\mathscr{D})$ returns the most recent dispute, $PreviousMoveType(\mathscr{D})$ denotes the type of the most recent move, and $Proposal(d)$ is d's root argument.

2.2.2 Moves

The applicability of dialogue moves is specified by sets of conditions and effects. The *propose* move initiates a new dispute. The agents are restricted to propose new arguments from $\mathcal{P}$, which is the set of all possible proposals.

$\mu_{l,0} = \langle propose, i, P \rangle$	
Conditions:	Effects:
$PreviousMoveType(\mathscr{D}) \in \{close, no\text{-}proposal\}$, or $\mathscr{D} = \langle\rangle$	Roles are switched
$\forall d$ in $\mathscr{D}$, $Proposal(d) \neq P$	$\mathcal{T}_{l,0} := \langle P \rangle$
$P \in GE_{\langle \mathscr{A}^i_{l-1} \cup \{P\}, \mathscr{R} \rangle}$	$V_{l,0} := P$
$P \in \mathcal{P}$	$CS^{PRO}_{l,0} := \{P\}$
	$L_{l,0} := \{P\}$
	$Q_{l,0} := \emptyset$
	$\mathscr{A}_{l,0} := \emptyset$

[1]This paper assumes two-player situations; in case of more than two agents, our results carry over assuming pairwise dialogues are conducted between all agents to reach agreement among the full set of agents.

The *no-proposal* move is made when an agent is unable to present a new proposal. Its conditions are: $PreviousMoveType(\mathscr{D}) = close$ or $\mathscr{D} = \langle\rangle$ and $\nexists P\ s.t.\ Legal(\mu, \mathscr{D})$, for $\mu = \langle propose, i, P\rangle$.

The *terminate* move can be used after a no-proposal move, in order to terminate the discussion when no alternative proposal can be presented by either one of the agents. The conditions for this move are: $PreviousMoveType(\mathscr{D}) = no\text{-}proposal$ and $\nexists P\ s.t.\ Legal(\mu, \mathscr{D})$, for $\mu = \langle propose, i, P\rangle$.

Counter moves respond to the other party's most recent argument in a dispute. The proponent cannot present conflicting arguments in the same dispute line.

$\mu_{l,k} = \langle counter, OPP, Y\rangle$	
Conditions:	Effects:
$PreviousMoveType(\mathscr{D}) \in \{propose, counter, retract\}$	$T_{l,k} := T_{k-1} + \langle Y, V_{k-1}\rangle$
$Y \in \mathscr{A}^{OPP}_{l-1} \cup CS^{PRO}_{l,k-1}$	$V_{l,k} := Y$
$Y \rightarrow V_{l,k-1}$	$CS^{PRO}_{l,k} := CS^{PRO}_{l,k-1}$
	$L_{l,k} := L_{l,k-1}$
	$Q_{l,k} := Q_{l,k-1}$
	$\mathscr{A}_{l,k} := \mathscr{A}_{l,k-1} \cup Y$

$\mu_{l,k} = \langle counter, PRO, Y\rangle$	
Conditions:	Effects:
$PreviousMoveType(\mathscr{D}) \in \{counter, backup\}$	$T_{l,k} := T_{k-1} + \langle Y, V_{k-1}\rangle$
$Y \in \mathscr{A}^{PRO}_{l-1}$	$V_{l,k} := Y$
$Y \rightarrow V_{l,k-1}$	$CS^{PRO}_{l,k} := CS^{PRO}_{l,k-1} \cup \{Y\}$
$Y \notin CS^{PRO}_{l,k-1}$	$L_{l,k} := L_{l,k-1} \cup \{Y\}$
	$Q_{l,k} := Q_{l,k-1}$
$CS^{PRO}_{l,k-1} \cup \{Y\}$ is conflict-free	$\mathscr{A}_{l,k} := \mathscr{A}_{l,k-1} \cup Y$
$\forall R \in Q_{l,k},\ R \not\subseteq L_{l,k-1} \cup \{Y\}$	

OPP may use the *backup* move to backtrack and provide an alternative defeat if she cannot counter *PRO*'s latest argument.

$\mu_{l,k} = \langle backup, OPP, Y, X, \delta\rangle$
Conditions:
$PreviousMoveType(\mathscr{D}) \in \{counter\}$
$Y \in \mathscr{A}^{OPP}_{l-1} \cup CS^{PRO}_{l,k-1}$
$X = V_b$ is the most recent argument in the dispute line $V_n \rightarrow \ldots \rightarrow V_b \rightarrow \ldots \rightarrow P$ for which:
$-\ \delta = Depth(X) + 1$ is odd
$-\ Y \rightarrow X$
$-\ B \notin Children(\mathcal{T}_{l,k}, X, Depth(X))$.
Effects:
$\mathcal{T}_{l,k+1} := \mathcal{T}_k + \langle Y, X\rangle$
$V_{l,k+1} := Y$
$CS^{PRO}_{l,k+1} := CS^{PRO}_{l,k-1} \setminus \{V_n, V_{n-2}, \ldots, V_{b+1}\}$
$L_{l,k} := L_{l,k-1}$
$Q_{l,k} := Q_{l,k-1}$
$\mathscr{A}_{l,0} := \emptyset$

The retract move can be used by the proponent in order to attempt to provide an alternative line of defence if it is not possible to counter an argument presented by *OPP*.

$\mu_{l,k} = \langle retract, PRO\rangle$	
Conditions:	Effects:
$PreviousMoveType(\mathscr{D}) \in \{counter, backup\}$	$\mathcal{T}_{l,k+1} := \langle P\rangle$
$\nexists X\ s.t.\ Legal(\mu, \mathscr{D})$ for	$V_{l,k+1} := P$
$\mu = \langle counter, PRO, X\rangle$	$CS^{PRO}_{l,k+1} := CS^{PRO}_{l,0}$
	$L_{l,k} := L_{l,0}$
	$Q_{l,k} := Q_{l,k-1} \cup \{L_{l,k-1}\}$
	$\mathscr{A}_{l,k} := \mathscr{A}_{l,k-1}$

The *accept proposal* move is preformed by the opponent, closing the most recent dispute as a *failing attack* and terminate the dialogue in favour of the most recent proposal.

$\mu_{l,k+1} = \langle accept, OPP\rangle$
Conditions:
$PreviousMoveType(\mathscr{D}) \in \{propose, counter\}$
$\nexists Y\ s.t.\ Legal(\mu, \mathscr{D})$, for $\mu = \langle counter, OPP, Y\rangle$
$\nexists Y, X, \delta\ s.t.\ Legal(\mu, \mathscr{D})$ for $\mu = \langle backup, OPP, Y, X, \delta\rangle$
Effects:
$\mathscr{A}^{PRO}_l := \mathscr{A}^{PRO}_{l-1} \cup \mathscr{A}_{l,k}$
$\mathscr{A}^{OPP}_l := \mathscr{A}^{OPP}_{l-1} \cup \mathscr{A}_{l,k}$
$Proposal(d_l)$ is accepted

The *close dispute* move is available to the proponent and closes the most recent dispute as a *failing defence*.

$\mu_{l,k+1} = \langle close, PRO\rangle$
Conditions:
$PreviousMoveType(\mathscr{D}) \in \{counter, backup\}$
$\nexists Y\ s.t.\ Legal(\mu, \mathscr{D})$, for $\mu = \langle counter, PRO, Y\rangle$
$\nexists Legal(\mu, \mathscr{D})$, for $\mu = \langle retract, PRO, Y, X, \delta\rangle$
Effects:
$\mathscr{A}^{PRO}_l := \mathscr{A}^{PRO}_{l-1} \cup \mathscr{A}_{l,k}$
$\mathscr{A}^{OPP}_l := \mathscr{A}^{OPP}_{l-1} \cup \mathscr{A}_{l,k}$

We consider a *strategy* to be a set of rules that select exactly one move from the set of all legal moves. The *confident* strategy constructs a move based on a complete ordering over all possible legal options. Preference over moves is calculated according to: (i) the following ordering over move types: $l = \langle counter, backup, retract, close, accept, propose, no\text{-}proposal, terminate\rangle$, (ii) the preference level of the argument presented by the move, for moves of the same type. If the preference ordering over arguments is partial, the agent constructs a complete ordering by randomly ordering the equally prefered moves.

2.3 Properties

In this section we will present important properties of the protocol. The following proofs assume that the dialogue is conducted between two agents with initially consistent, finite argument sets, following confident strategies.

PROPOSITION 1. *(Termination) An iterated dispute for agents with finite argument sets always terminates.*

PROOF. The proponent cannot repeat the same arguments in the same dispute line, and cannot repeat infinite *backup* moves as each one represents an alternative line of defence. The agents' arguments are finite. Therefore, dispute will always terminate. If there exists a dispute that is a failing attack of the proposal, the iterated dispute will terminate. We show that there can be no infinite sequence of disputes that are all failing defences. For proposal P and dispute $l+1$, if d_{l+1} is a failing defence of P, there exists a set of arguments *OPP* against which P cannot be defended. *PRO* can only present proposals that are part of $GE_{\langle \mathscr{A}^{PRO}_l \cup \{P\}, \mathscr{R}\rangle}$, which are defended against all defeats from $\mathscr{A}^{PRO}_l$. Since P is not defended against all defeats in the dispute, there exists at least one argument B that was presented by *OPP* and is not part of $\mathscr{A}^{PRO}_l$. The agents have finite argument sets and after every dispute they learn the arguments presented by the other party. So there cannot be an infinite sequence of disputes that are failing attacks. Therefore, an iterated dispute always terminates. □

In order to prove soundness, we introduce two key lemmas. The following lemma shows that, if a dispute d_l is a failing attack of a proposal P, then P will be in the grounded extension of the argumentation framework $\langle \mathscr{A}^{OPP}_l \cup \{P\}, \mathscr{R}\rangle$, where $\mathscr{A}^{OPP}_l$ are the arguments that the opponent will know after the the dispute terminates.

LEMMA 1. *Let d_l be a closed dispute about P between agents PRO and OPP, with initially consistent finite argument sets $\mathscr{A}_0^{PRO} = \mathcal{A}$ and $\mathscr{A}_0^{OPP} = \mathcal{B}$ that follow a confident strategy. If d_l is a failing attack of P then $P \in GE_{AF_l^{OPP}}$, for the argumentation framework $AF_l^{OPP} = \langle \mathscr{A}_l^{OPP} \cup \{P\}, \mathscr{R} \rangle$, where $\mathscr{A}_l^{OPP}$ denotes the arguments known to OPP after the dispute d_l.*

PROOF. All nodes presented by OPP in the dispute tree $\mathcal{T} = \mathcal{T}_{l,k}$ have one child node presented by PRO, since the proponent can only add a new argument to the dispute tree using the counter move. A dispute closes as a failing attack if the proponent counters every counter and backup move made by the opponent. Therefore, all leaf nodes are presented by PRO. We will show that all arguments of even depth in $\mathcal{T}$ are part of the grounded extension of AF_l^{OPP} by induction over the distance between them and the leaf nodes in the tree.

(Base Case) For *distance* $= 0$, let $\mathcal{V}_0$ be the leaf node arguments of depth n. These arguments were presented by PRO. Also, $\forall A_n \in \mathcal{V}_0, \nexists B_{n-1} \in \mathscr{A}_{OPP} \cup CS_{PRO}$ *s.t.* $B_{n-1} \rightarrow A_n$, because if such an argument existed OPP would have presented it, due to the specification of the confident strategy and the counter and backup moves. Let $\mathscr{F}^i_{OPP}$ be the characteristic function of AF_l^{OPP}. All leaf node arguments in $\mathcal{V}_0$ are part of $\mathscr{F}^1_{OPP}$, since there are no arguments in $\mathscr{A}_l^{OPP}$ defeating them. So $\mathcal{V}_0 \in GE_{AF_l^{OPP}}$.

(Induction Step) We assume that the property holds for arguments of distance k from the leaf nodes, $\mathcal{V}_k \subseteq GE_{AF_l^{OPP}}$. We will show that the property holds for arguments of distance $k+2$. All arguments of distance $k+2$ from the leaf node $\mathcal{V}_{k+2}$ are defeated by an argument in $\mathcal{V}_{k+1}$, which are in turn defeated by arguments in $\mathcal{V}_k$. Also, there is no other $B'_{k+1} \in \mathscr{A}_l^{OPP}$ that defeats any argument in $\mathcal{V}_{k+2}$ that has not been presented, because of OPP's strategy. According to the induction step $\mathcal{V}_k \subseteq GE_{AF_{OPP}}$, so $\mathcal{V}_{k+2} \subseteq GE_{AF_l^{OPP}}$.

Therefore, all arguments of even distance from the leaf nodes will be part of $GE_{AF_l^{OPP}}$, including P. $\square$

The following lemma shows that for agents with finite and conflict-free initial argument sets, which have exchanged subsets of their arguments, if a proposal is acceptable with respect to the arguments both agents know, then it will be also acceptable with respect to both agents' arguments.

LEMMA 2. *Let $AF_1 = \langle \{P\} \cup \mathcal{A} \cup \mathcal{B}', \mathscr{R} \rangle$, $AF_2 = \langle \{P\} \cup \mathcal{A}' \cup \mathcal{B}, \mathscr{R} \rangle$ and $AF = \langle \{P\} \cup \mathcal{A} \cup \mathcal{B}, \mathscr{R} \rangle$, with $\mathcal{A}$, $\mathcal{B}$ finite, conflict-free argument sets, $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{B}$. If $P \in GE_{AF_1} \cap GE_{AF_2}$ then $P \in GE_{AF}$.*

PROOF. We will first show by induction on the characteristic function $\mathscr{F}_{AF_1}$ that $\forall B \in \mathcal{B}'$ *s.t.* $B \in GE_{AF_1}$ it holds that $B \in GE_{AF}$. It holds that $\mathcal{B}$ is conflict-free, and $B, P \in GE_{AF_1}$, therefore all arguments defeating B will be in $\mathcal{A}$. (Base case) $\forall B \in \mathscr{F}^1_{AF_1}$, there exists no argument defeating B in $\{P\} \cup \mathcal{A} \cup \mathcal{B}'$. There will be no argument defeating B in $\{P\} \cup \mathcal{A} \cup \mathcal{B}$, so $B \in GE_{AF}$. (Induction step) We assume that $\forall B \in \mathscr{F}_{AF_k}$, $B \in GE_{AF}$, and we show that is holds for $\forall B \in \mathscr{F}_{AF_{k+1}}$. All arguments attacking B in $\{P\} \cup \mathcal{A} \cup \mathcal{B}'$ are also part of $\{P\} \cup \mathcal{A} \cup \mathcal{B}$. Since $B \in \mathscr{F}_{AF_{k+1}}$ it is defended against these attacks by arguments in $\mathscr{F}_{AF_k}$. These arguments are part of GE_{AF} according to the induction step. Therefore, B will also be in GE_{AF}.

$P \in GE_{AF_1}$, therefore for all A in $\mathcal{A}$ defeating P, there is some $B \in GE_{AF_1}$ s.t. B defeats A. B will also be in GE_{AF}. Therefore, for any argument $A \in \mathcal{A}$ defeating P, there will exist $B \in GE_{AF}$ defeating A.

Accordingly, we can show that for any $B \in \mathcal{B}$ defeating P, there exists an argument $A \in GE_{AF}$ defeating B. Therefore, B is defended against all defeats from $\{P\} \cup \mathcal{A} \cup \mathcal{B}$ by arguments in GE_{AF}. So $P \in GE_{AF}$. $\square$

The following proposition asserts that for agents following confident strategies, with initially conflict-free argument sets, if a proposal is accepted by the dialogue, then it is acceptable with respect to the union of their arguments.

PROPOSITION 2. *(Soundness) If an iterated dispute between two agents i, j following confident strategies, terminates accepting a proposal argument P, then P is in the grounded extension of the argumentation framework $AF = \langle \mathscr{A}_1 \cup \mathscr{A}_2 \cup \{P\}, \mathscr{R} \rangle$, where $\mathcal{A}$ and $\mathcal{B}$ are the initial, finite and conflict-free argument sets for agent i and j respectively.*

PROOF. Let i be the agent that made the accepted proposal. Consider the following argumentation frameworks: $AF^{PRO} = \langle \mathcal{A} \cup \mathcal{B}' \cup \{P\}, \mathscr{R} \rangle$ and $AF^{OPP} = \langle \mathcal{B} \cup \mathcal{A}' \cup \{P\}, \mathscr{R} \rangle$. $\mathcal{A} \cup \mathcal{B}'$ denotes the arguments PRO knew before initiating the final dispute and $\mathcal{B} \cup \mathcal{A}'$ is the set of all the arguments OPP knows after the dispute has terminated. $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{B}$. The agents follow confident strategies, so the proponent will propose arguments that are in the grounded extension of AF^{PRO}. According to Lemma 1 the proposal argument will be in the grounded extension of the opponent $GE_{AF^{OPP}}$ if a dispute terminates as a failing attack. According to Lemma 2 if the proposed argument is in $GE_{AF^{PRO}} \cap GE_{AF^{OPP}}$ then it is part of GE_{AF}. $\square$

3. ARGUMENTS ABOUT PLANS

Given that we now have a working protocol for iterated dispute dialogues, we can now introduce a logic for arguments about plans that is suitable for combination with contemporary AI planning systems.

3.1 Situation Calculus

Situation calculus is a language for the representation of dynamic domains [8]. It supports three disjoint sorts. The sort *action* represents actions, the sort *situation* represents situations (i.e. histories of action sequences) and the sort *object* all the rest. S_0 is a constant symbol representing the initial situation. The binary function symbol $do : action \times situation \rightarrow situation$ denotes the successor situation after performing an action. $Poss : action \times situation$ is a binary predicate symbol representing whether an action is applicable in a situation. The binary predicate symbol $\sqsubset$: $situation \times situation$ defines an ordering relation over situations, where $s \sqsubset s'$ denotes that s is a proper subsequence of s'. Symbols whose value change in different situations are called fluents (relational or functional), and they have an argument of sort *situation* as their final argument.

Reasoning about dynamic domains can be performed in structured situation calculus theories called *basic action theories* in a way that overcomes the frame problem [11]. A basic action theory $\mathcal{D}$ has the following form:

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0}$$

Σ is a set of fundamental domain-independent axioms providing the basic properties for situations. $\mathcal{D}_{ss}$ contains a successor state axiom for each relational fluent in the domain, which specifies all the conditions that govern its value. The conditions under which an action can be performed are specified by the action precondition axioms $\mathcal{D}_{ap}$. $\mathcal{D}_{una}$ contains the unique names axioms for actions. $\mathcal{D}_{S_0}$ is a set of first-order sentences that represent the initial state of the world. Action Precondition axioms specify the preconditions of an action in a first-order statement: $\forall \vec{x}, s \;.\; Poss(A(\vec{x}), s) \equiv \Pi_A(\vec{x}, s)$. Each Successor State axiom describes the conditions that should hold in a situation, in order for a fluent to take on a specific value after performing an action. Axioms for relational fluents have the following form: $\forall \vec{x}, a, s \;.\; F(\vec{x}, do(a, s)) \equiv \Phi_F(\vec{x}, a, s)$ (where $\Phi_F(\vec{x}, a, s)$ describes the necessary previous situation conditions). Let *Fluents*/*NonFluents* be the set of all fluent/non-fluent symbols in the domain and *Actions* the set of all actions available to the agents. $G(s)$ denotes a shared goal. We assume that the sets *Fluents*, *NonFluents* and *Actions* are shared among the agents. Also, we assume that agents share the planning goal, knowledge about the fundamental axioms, unique names axioms for actions and the names of the objects in the domain.

3.2 Arguments

The arguments held by the agents contain statements in the language of situation calculus. We will define arguments based on an inference procedure $\vdash$ in a knowledge base, and reinterpret the defeat relation using logical contradiction:

Definition 4. Arguments *for agent* $i \in \{1, 2\}$ *are pairs* $A = \langle H, h\rangle$, *where* $H \subseteq 2^{\mathcal{L}_{SitCal}}$, *where* $\mathcal{L}_{SitCal}$ *is the set of all well-formed situation calculus sentences.*

i. *H is consistent (i.e. $H \nvdash \bot$),*

ii. $H \vdash h$,

iii. *H is minimal (no subset of H satisfies both* i. *and* ii.*).*

H is called the support of the argument and h its conclusion. We will also use the following notation: $Support(A) = H$, and $Claim(A) = h$. The preference level of an argument A is denoted by $pLevel(A)$, and is the minimum preference level of a statement in $Support(A)$.

Definition 5. *An argument $A_1 = \langle H_1, h_1\rangle$* defeats *an argument $A_2 = \langle H_2, h_2\rangle$, denoted $A_1 \to A_2$, if $pLevel(A_1) \geq pLevel(A_2)$ and, either there $\exists\phi$ in $H_2 \cup \{h_2\}$ such that $h_1 \equiv \neg\phi$ or $h_1 = (F \equiv \Psi)$, $\phi = (F \equiv \Phi)$ and $\Phi \neq \Psi$, where F is a predicate symbol.*

The defeats relation considers contradictory beliefs and formulas providing different definitions of the same symbol. Reasoning about actions is based on the axioms representing the domain. It is essential that the domain theory does not include different axioms regarding the same predicate.

3.3 Planning Knowledge

The *domain knowledge* for agent i after dispute k, is a set of beliefs $\mathcal{B}^i_k$, generated using the arguments in the grounded extension of AF^i_k, $GE_{AF^i_k}$, as illustrated below:

Compute $GE_{AF^i_k}$, for $AF^i_k := \langle \mathscr{A}^i_k, \mathscr{R}\rangle$;
$\mathcal{B}^i_k := \{\phi \mid V \in GE_{AF^i_k} \wedge Claim(V) = \phi\}$;
forall $\phi \in \mathcal{B}^i_k$ **do** $pLevel(\phi) := \max_{V \in GE_{AF^i_k}} pLevel(V)$;
return $\mathcal{B}^i_k$;

We consider the initial argument sets of the agents to be conflict-free. If all of their claims are basic action theory statements, all future domain beliefs constructed by this algorithm will be basic action theories.

3.4 Proposals

A plan for a goal G is represented in situation calculus by the statement: $executable(S_\pi) \wedge G(S_\pi)$. S_π is a variable-free situation term representing the history for the execution of the actions of the plan in sequence, and $executable(s_\pi) \stackrel{\text{def}}{=} (\forall a, s^*).do(a, s^*) \sqsubseteq s_\pi \supset Poss(a, s^*)$. A consequence of the definition of a plan and the foundational axioms for situations is that $\forall a, s \;.\; executable(do(a, s)) \equiv executable(s) \wedge Poss(a, s)$.

Definition 6. *Plan proposal arguments for agent i after dispute k for a shared goal G, are all arguments P s.t. i) $Claim(P) = G(S_\pi) \wedge executable(S_\pi)$, with $S_0 \sqsubset S_\pi$, ii) $Support(P)$ is the minimal subset of $\mathcal{B}^i_k$ s.t. $Support(P) \vdash Claim(P)$.*

If P is a plan proposal argument then $P \in \mathcal{P}$. The preference level of a plan proposal argument is equal to the lowest preference level of the claims in its support.

The agents can obtain the support set of a plan proposal argument using Reiter's *regression operator* $\mathcal{R}$. The regression operator eliminates statements with complex situation terms by replacing them with logically equivalent statements that refer to situations closer to the initial state. The process is repeated until all fluents in the statement refer to the initial situation. The logical equivalence follows from the relevant action preconditions and the successor state axioms. A detailed analysis of the regression operator can be found in [11].

The support of a proposal argument contains the minimal subset of domain beliefs and unique names assumptions sufficient to infer $\mathcal{R}[Claim(A)]$, as well as the equivalencies that were employed by the regression operator.

Example 1. *Using the axiom $\forall a, s \;.\; f(do(a, s)) \equiv (a = A_1) \vee (a = A_2) \vee (f(s) \wedge a \neq A_3)$, $\mathcal{R}[f(do(A_4, S_0))]$ returns $A_4 = A_1 \vee A_4 = A_2 \vee f(S0) \wedge A_4 \neq A_3$, which can be simplified to $f(S_0)$.*

The following proposition asserts that for all plans that can be constructed from an agent's planning knowledge, the plan proposal arguments for these plans will be part of the grounded extension of $\langle \mathscr{A}^i_l \cup \{P\}, \mathscr{R}\rangle$, where $\mathscr{A}^i_l$ are the arguments known to agent i after dispute d_l.

Proposition 3. *If P is a plan proposal argument with $Claim(P) = G(S_\pi) \wedge executable(S_\pi)$, constructed in iteration k by agent i, then $P \in GE_{\langle \mathscr{A}^i_l \cup \{P\}, \mathscr{R}\rangle}$.*

Proof. P does not defeat any argument in $GE_{\langle \mathscr{A}, \mathscr{R}\rangle}$, since $Claim(P)$ refers to a future situation S_π with $S_0 \sqsubset S_\pi$, whereas all statements in the claim or support of arguments in $GE_{\langle \mathscr{A}, \mathscr{R}\rangle}$ are initial situation statements, non-fluent statements and domain axioms. Therefore, $\forall V \in GE_{\langle \mathscr{A}^i_l, \mathscr{R}\rangle}$, it will be the case that $V \in GE_{\langle \mathscr{A}^i_l \cup \{P\}, \mathscr{R}\rangle}$. All statements in the support of P are claims of arguments in $GE_{\langle \mathscr{A}^i_l \cup \{P\}, \mathscr{R}\rangle}$, therefore any defeats against P will be defended by arguments in the grounded extension. So $P \in GE_{\langle \mathscr{A}^i_l \cup \{P\}, \mathscr{R}\rangle}$. □

Proposition 4 extends the result of Proposition 2 to dialogues about plans.

PROPOSITION 4. *If an iterated dispute terminates with both agents accepting a plan proposal P, then $P \in GE_{\langle \mathscr{A} \cup P, \mathscr{R} \rangle}$, for agents with initially conflict-free argument sets following confident strategies.*

Proof follows from propositions 2 and 3. This proposition asserts that under the aforementioned assumptions, if a plan is proposed by the proponent and accepted by the opponent though the dialogue, then this plan is acceptable with respect to the union of the arguments of all agents.

Plan generation is conducted as planning. Planning can be performed in situation calculus [11], or by an equivalent PDDL representation [12] using a standard planner.

3.5 Example

An informal example of a run of an iterated dispute about plans is depicted in Figure 1. Two cooperative agents i and j share the goal G. The agents need a plan that achieves G. They will accept only plans that follow their beliefs.

Move	Dispute
$\langle propose, i, P_1 \rangle$ for plan $\pi_1 = \langle a_1, a_2 \rangle$	d_1
$\langle counter, j, V_1 \rangle$ with $Claim(V_1) = \neg\phi(S_0)$	
$\langle counter, i, V_2 \rangle$	
$\langle counter, j, V_3 \rangle$	
⋮	
$\langle close, i \rangle$	
$\langle propose, j, P_2 \rangle$ for plan $\pi_2 = \langle a_1, a_4, a_5 \rangle$	d_2
$\langle counter, i, V_4 \rangle$ with $Claim(V_4) = axiom_{a_5}$	
$\langle close, j \rangle$	
$\langle propose, i, P_3 \rangle$ for plan $\pi_3 = \langle a_1, a_7, a_5 \rangle$ supported by $\neg\phi(S_0)$	d_3
$\langle accept, j \rangle$	

Figure 1: Example of an iterated dispute

Agent i initiates the iterated dispute with the dispute d_1, proposing P_1 for the plan $\pi_1 = \langle a_1, a_2 \rangle$, with $\phi \in Support(P_1)$. Agent j counters the proposal using V_1 claiming $\neg\phi(S_0)$. A sequence of arguments is presented by both agents. Eventually, agent i is unable to defend the plan against a defeat, and the dispute closes as a failing defence.

Dispute d_2 is then initiated by j playing the role of the proponent. Agent j proposes P_2 presenting an alternative plan $\pi_2 = \langle a_1, a_4, a_5 \rangle$. Agent i counters the proposal using argument V_4, claiming $axiom_{a_5}$. The formula $axiom_{a_5}$ is a different action precondition axiom from the one used by j. The proponent cannot defend P_2 and closes the dispute.

Agents i continues the dialogue by initiating the dispute d_3, which proposes P_3 for $\pi_3 = \langle a_1, a_7, a_5 \rangle$. This plan is accepted by j. One of the beliefs that support P_3 is $\neg\phi(S_0)$. Agent i did not initially believe $\neg\phi(S_0)$, but its compliment. The belief was updated when i learned the argument V_1 claiming $\neg\phi(S_0)$ from j.

4. EXTENSIONS

Since plans have a much more restricted structure than general logical theories, we can design algorithms for argument and proposal generation specialised for disputes about plans. Some general extensions for guiding overall agent reasoning and reducing the size of plan proposal arguments are presented in the following sections.

4.1 Completing Domain Knowledge

Our assumptions do not guarantee that the domain knowledge of the agents is complete. Consider the following:

EXAMPLE 2. *Let $\mathscr{A}_0^i = \{A_0 : \langle \{b, b \rightarrow a\}, a \rangle,\ A_2 : \langle \{b\}, b \rangle\}$, the argument $B_1 : \langle \{\neg b\}, \neg b \rangle$ used by agent j during first dispute, with $pLevel(B_1) > pLevel(A_1)$ and $pLevel(B_1) > pLevel(A_2)$. $\mathscr{A}_1^j = \{A_0, A_1, B_1\}$, $GE^i_{\langle \mathscr{A}_1^j, \mathscr{R} \rangle} = \{B_1\}$ and $\mathcal{B}_1^i = \{\neg b\}$. The knowledge for agent i is not complete after one dispute, since he does not believe neither a or $\neg a$.*

Incomplete information about the domain knowledge complicates the reasoning process as plans must achieve the goal in every domain that follows the incomplete specification. Planning with incomplete information can be performed as open world planning [5], or conformant planning [13]. In this section, we will describe an algorithm for the construction of complete knowledge bases, which enables plan proposal generation through planning with complete information.

We assume that the claims of the agents' arguments have one of the following restricted forms:

i. $F(\vec{x})$ or $\neg F(\vec{x})$ with $F(\vec{x}) \in NonFluents$

ii. $F(\vec{x}, S_0)$ or $\neg F(\vec{x}, S_0)$ with $F \in Fluents$

iii. $\forall \vec{x}, s\ .\ Poss(A(\vec{x}), s) \equiv \Pi(\vec{x}, s)$

iv. $\forall \vec{x}, a, s\ .\ F(\vec{x}, do(a, s)) \equiv \Phi(\vec{x}, s)$

The domain knowledge is *complete* if for every non-fluent symbol $f \in NonFluents$ there is a statement of the form (i), for every fluent symbol $F \in Fluents$ there is a statement of the form (ii) and a definition of the form (iv), and for every action $A \in Actions$, there is a definition of the form (iii).

The following algorithm constructs the complete domain knowledge for agent i after dialogue l:

Construct $GE := GE_{\langle \mathscr{A}_l^i, \mathscr{R} \rangle}$;
Construct $\mathcal{B}_l^i$;
$\bar{\mathcal{B}}_l^i := \mathcal{B}_l^i$;
forall $\phi \in \mathcal{B}_0^i$ **do**
  if *ϕ is a successor state axiom for the fluent symbol F and there is no successor state axiom for F in $\bar{\mathcal{B}}_l^i$* **then**
    $\bar{\mathcal{B}}_l^i := \bar{\mathcal{B}}_l^i \cup \{\phi\}$;
  if *ϕ is an action precondition axiom for action A and there is no successor state axiom for A in $\bar{\mathcal{B}}_l^i$* **then**
    $\bar{\mathcal{B}}_l^i := \bar{\mathcal{B}}_l^i \cup \{\phi\}$;
  else if $\phi \notin \bar{\mathcal{B}}_k^i$ *and* $\hat{\phi} \notin \bar{\mathcal{B}}_l^i$ **then**
    $\bar{\mathcal{B}}_l^i := \bar{\mathcal{B}}_l^i \cup \{\phi\}$;
return $\bar{\mathcal{B}}_k^i$;

PROPOSITION 5. *If the claims of the argument sets of both agents follow the restricted forms, and the initial domain knowledge is complete, then for any l $\bar{\mathcal{B}}_l^i$ will be complete.*

Proof follows from the above algorithm as for any piece of missing knowledge the related piece of information is added from initial domain knowledge, asserting that if $\mathcal{B}_0^i$ is complete, then $\bar{\mathcal{B}}_k^i$ will be complete for any k.

The above algorithm does not assert that $P \in GE_{\langle \mathscr{A}_l^i \cup \{P\}, \mathscr{R} \rangle}$ before presenting the plan proposal argument. If $P \notin GE_{\langle \mathscr{A}_l^i \cup \{P\}, \mathscr{R} \rangle}$, then the agent must re-plan in order to find an alternative plan.

The following proposition asserts that if the grounded extension of the argumentation framework $\langle \mathscr{A}_l^i, \mathscr{R} \rangle$ is a stable extension, then any proposal arguments for plans constructed using $\bar{\mathcal{B}}_l^i$ will be in the grounded extension of

$\langle \mathscr{A}_l^i \cup \{P\}, \mathscr{R}\rangle$. *Stable* extensions are conflict-free sets that defeat[2] every argument that does not belong to it.

PROPOSITION 6. *Assuming that for all agents i, $\mathscr{A}_0^i$ is conflict-free, and $GE_{\langle \mathscr{A}_l^i, \mathscr{R}\rangle}$ is a stable extension, if $\bar{B}_l^i \vdash G(s) \wedge executable(s)$, then the proposal argument P for the related plan will be in $GE_{\langle \mathscr{A}_l^i \cup \{P\}, \mathscr{R}\rangle}$.*

PROOF. For any $\phi \in B_l^i$, there will be an argument V, with $Claim(V) = \phi$ which is part of $GE_{\langle \mathscr{A}_l^i, \mathscr{R}\rangle}$ (see proof of Proposition 3). For any $\phi \in \bar{B}_l^i \setminus B_l^i$ $\nexists V \in GE_{\langle \mathscr{A}_l^i, \mathscr{R}\rangle}$ s.t. $Claim(\mathscr{A}_l^i)$ is the compliment of ϕ (if ϕ is a fluent or non-fluent statement), or an axiom about the same fluent/action (if ϕ is a successor state axiom/ action precondition axiom), and since $GE_{\langle \mathscr{A}_l^i, \mathscr{R}\rangle}$ is a stable extension, there will be some $V' \in GE_{\langle \mathscr{A}_l^i, \mathscr{R}\rangle}$ s.t. $V' \rightarrow V$. P does not attack any argument in $\mathscr{A}_l^i$, therefore $\forall V \in GE_{\langle \mathscr{A}_l^i, \mathscr{R}\rangle}$, it holds that $V \in GE_{\langle \mathscr{A}_l^i \cup \{P\}, \mathscr{R}\rangle}$. Therefore, $\forall V\ s.t.\ V \rightarrow P$, $\exists V' \in GE_{\langle \mathscr{A}_l^i \cup \{P\}, \mathscr{R}\rangle}$ s.t. $V' \rightarrow V$. Therefore, $P \in GE_{\langle \mathscr{A}_l^i \cup \{P\}, \mathscr{R}\rangle}$. $\square$

This provides us with clear criteria for cases in which all plans constructed through planning with complete information are acceptable to the proponent and can be directly proposed in the dialogue.

4.2 Minimal Plan Proposals

The support of a plan proposal argument is the minimal set of domain beliefs from which the claim of the proposal argument can be deduced. Depending on the length of the plan and the form of the axioms, the size of the support set can be extensive. In the worst case it can be comparable in size to the entire domain knowledge. In this section we will present an alternative form of plan proposal arguments and discuss the advantages and drawbacks of such an approach.

Let a *minimal plan proposal argument* P be an argument with $Claim(P) = G(S_\pi) \wedge executable(S_\pi)$ and $Support(P) = \{G(S_\pi), executable(S_\pi)\}$. If $\mathcal{B}_l^{PRO}$ are the beliefs for agent i after iteration l, then $\forall \phi \in Support(P)$, $\mathcal{B}_l^{PRO} \vdash \phi$.

Minimal plan proposal arguments cannot be defeated by any argument in $\mathscr{A}$, as their support contains only statements about a future situation. We extend our protocol, enabling the opponent to challenge the support of minimal proposal arguments, and the proponent to expand them accordingly. The *challenge* move challenges a future situation statement in the support of a plan proposal argument.

$\boldsymbol{\mu_{l,k} = \langle challenge, OPP, \phi\rangle}$	
Conditions:	Effects:
$PreviousMoveType(\mathscr{D}) \in \{proposal, expand, counter\}$ $\mathcal{T}_k = \langle P\rangle$ $\exists \phi \in Support(P)$, which mentions S, with $S_0 \sqsubset S$ and $\mathcal{B}_l^{OPP} \nvdash \phi$	$\mathcal{T}_{l,k} := \mathcal{T}_{l,k-1}$ $V_{l,k} := null$ $CS_{l,k}^{PRO} := CS_{l,k-1}^{PRO}$ $P_{l,k} := P_{l,k-1}$ $Q_{l,k} := Q_{l,k-1}$ $\mathscr{A}_{l,k} := \mathscr{A}_{l,k-1}$

The *expand* move can be used after a challenge move, extending the proposal argument and justifying the challenged support. This move works as one-step regression.

The confident strategy is extended accordingly with $l = \langle$*counter*, *backup*, *retract*, *challenge*, *expand*, *close*, *accept*, *propose*, *no-proposal*, *terminate*$\rangle$.

[2]A set defeats an argument if there is an argument in the set that defeats this argument.

$\boldsymbol{\mu_{l,k} = \langle expand, PRO, \Phi\rangle}$	
Conditions:	Effects:
$\mu_k = \langle challenge, OPP, \phi\rangle$ Φ is consistent $\Phi \vdash \phi$	$\mathcal{T}_{l,k} := \mathcal{T}_{l,k-1}$ with P replaced by P' $V_{l,k} := P'$ $CS_{l,k}^{PRO} := (CS_{l,k-1}^{PRO} \setminus \{P\}) \cup \{P'\}$ $P_{l,k} := (P_{l,k-1} \setminus \{P\}) \cup \{P'\}$ $Q_{l,k} := \emptyset$ $\mathscr{A}_{l,k} := \emptyset$ Where: – $Claim(P') := Claim(P)$ – $Support(P') := (Support(P) \setminus \{\phi\}) \cup \Phi$

Figure 2 illustrates an example of how these moves affect the support of a plan proposal argument. The different rows in the trapezoid illustrate different depths of support for the proposal. The minimal proposal argument is only supported by $\phi(S_2)$, whereas the complete support for the proposal argument is the set $\{axiom_\phi, \phi(S_0), \psi(S_0), \chi(S_0), axiom_\psi\}$. V_1 and V_2 are the opponent's argument defeating the proposal argument. The first defeat against the plan can be presented by the opponent, after the minimal proposal argument has been expanded to include a belief which can be defeated by arguments V_1 or V_2. If $\phi(S_2)$ is challenged, the proponent expands the proposal by replacing it with the formulas $\{axiom_\phi, \phi(S_1), \psi(S_1)\}$, which are formulas sufficient to prove $\phi(S_2)$. The opponent can defeat P using V_1. If the proponent cannot defend the proposal against this attack, then the dispute closes as a failing defence. In this case the formulas that were communicated by *PRO* are $\{\phi(S_2), axiom_\phi, \phi(S_1), \psi(S_1)\}$.

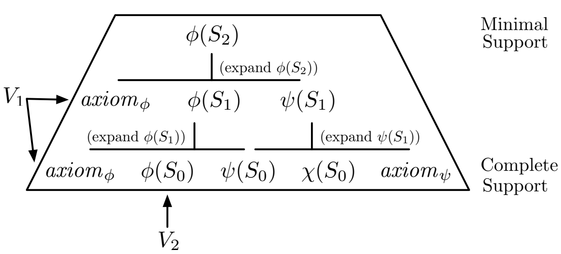

Figure 2: Example of the expansion of the support of a plan proposal argument

A line of consecutive challenges and expansions can be continued until the support of the proposal includes statements about the initial state. Each expansion replaces a sentence with a set of axioms and sentences of the previous situation that are sufficient to derive it. The number of consecutive challenges and expansions is bounded by the number of actions in the plan. The opponent can terminate a line of challenges and expansions using a counter move. If the proponent does not defend the plan against all defeats, then the dispute terminates, and there is no need for the proponent to communicate the remaining support of P.

Continuing the previous example, let us assume that the agents have equal probabilities for winning sub-disputes V_1 and V_2. *OPP* has double the chance of successfully defeating P using V_1 and V_2 than *PRO* to defend all defeats, thus making more probable the case that *PRO* will not have to communicate $\chi(S_0)$ and $axiom(\psi)$.

This extension is useful for proposals with extensive support. The benefit is that the proponent will need to expand the support of P only for the statements for which the agents disagree. An extreme case in which this extension minimises the required communication is when the opponent immediately agrees with the claim of the plan proposal argument, without making any challenges or counter moves.

The modified protocol does not always produce sound results with respect to the union of both agents' arguments. The following propositional example illustrates this issue:

EXAMPLE 3. *Let* $\mathcal{B}^i = \{b, b \rightarrow a, \neg c\}$ *and* $\mathcal{B}^j = \{\neg b, c \rightarrow a, c\}$*, with* $pLevel(\langle\{\neg c\}, \neg c\rangle) > pLevel(\langle\{c\}, c\rangle)$ *and* $pLevel(\langle\{\neg b\}, \neg b\rangle) > pLevel(\langle\{b\}, b\rangle)$*. Both agents will not challenge and cannot counter the minimal proposal argument* $\langle a, \{a\}\rangle$*, since* a *can be derived from both agents' beliefs. However, both related complete proposal arguments* $\langle a, \{b, b \rightarrow a\}\rangle$ *and* $\langle a, \{c, c \rightarrow a\}\rangle$*, are not acceptable w.r.t. the union of both agents' argument sets.*

The modified version can substantially reduce the size of the proposal arguments, when agents disagree about a small subset of their knowledge. This protocol produces sound results w.r.t. both argumentation frameworks $\langle \mathscr{A}_{l-1}^{PRO} \cup \{P\}, \mathscr{R}\rangle$, and $\langle \mathscr{A}_{l}^{OPP} \cup \{P\}, \mathscr{R}\rangle$, for a minimal proposal P.

5. CONCLUSION AND RELATED WORK

In this paper we define an abstract argumentation-based dialogue protocol that enables agents to discuss different proposals and agree on options that are acceptable w.r.t. their collective knowledge. The protocol searches the space of proposals iteratively. As the discussion progresses, the individual argument sets of the agents gradually converge. We extend the protocol for the specific problem of arguing about plans, and suggest extensions that guide agent reasoning and reduce the size of the proposal arguments. Analytical proofs are provided for important properties of our methods.

Our approach is influenced by recent work on argumentation for practical reasoning and deliberation [1, 10, 14] and planning over defeasible knowledge [6]. We maintain a closer relation to classical AI planning, instead of allowing very expressive dialogues about goals and intentions of agents. We employ a standard formalism for the representation of the planning domain which can be transformed to the ADL fragment of PDDL [12].

Our methods are related to multiagent planning [9], but solve a different problem. Multiagent planning usually refers to distributed planning, distributed execution, or both. Multiagent planning problems include distributing the planning process in order to improve efficiency, or coordinating plans constructed by different agents to avoid unwanted interference among their actions. On the other hand, our framework utilises the agents' domain knowledge in order to align inconsistent beliefs and identify plans that will be accepted by all agents. Our system treats plans as fully ordered sequences of actions. Plans can be joint or single agent plans. Multiagent dialogue searches the plan space for an acceptable plan. The actual planning process is single-agent.

Our work is also related to planning under uncertainty and to conformant planning, as for instance [13], although the problem in this case is different. Conformant plans achieve the goal in all worlds that result from the combination of the uncertain pieces of information about the planning domain. Instead, our system deals with plans that need to satisfy n different world views (one view for each planning agent). In addition, our system allows persuasion, enabling the acceptance of plans that initially seemed unacceptable.

Other relevant approaches with different focus include frameworks that employ argumentation to reason about dynamic domains [7, 15]. Our system uses argumentation-based dialogue for resolving conflicting beliefs of different planning agents. Reasoning about change is treated in a monotonic manner in our framework.

In the future we would like to look into restricted versions of the language for which we can guarantee complete results for a computationally tractable protocol. In addition, we would like to work on heuristics guiding the search based on specific attributes of the planning problem.

6. REFERENCES

[1] K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10-15):855–874, 2007.

[2] A. Belesiotis, M. Rovatsos, and I. Rahwan. A generative dialogue system for arguing about plans in situation calculus. In *Proceedings of ArgMAS-09*, May 2009.

[3] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[4] P. E. Dunne and T. Bench-Capon. Two party immediate response disputes: properties and efficiency. *Artificial Intelligence*, 149(2):221–250, 2003.

[5] A. Finzi, F. Pirri, and R. Reiter. Open world planning in the situation calculus. In *Proceedings of AAAI-2000*, pages 754–760. AAAI Press, 2000.

[6] D. Garcia, A. Garcia, and G. Simari. Planning and defeasible reasoning. In *Proceedings of AAMAS-07*, 2007.

[7] A. Kakas, R. Miller, and F. Toni. An argumentation framework for reasoning about actions and changes. *Lecture notes in computer science*, pages 78–91, 1999.

[8] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4, 1969.

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

[10] I. Rahwan and L. Amgoud. An argumentation based approach for practical reasoning. In *Proceedings of AAMAS-06*, pages 347–354, 2006.

[11] R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[12] G. Röger, M. Helmert, and B. Nebel. On the relative expressiveness of ADL and Golog: The last piece in the puzzle. In *Proceedings of KR-08*, 2008.

[13] D. Smith and D. Weld. Conformant graphplan. In *Proceedings of AAAI-98*, pages 889–896, 1998.

[14] Y. Tang and S. Parsons. Argumentation-based dialogues for deliberation. In *Proceedings of AAMAS-05*, pages 552–559, 2005.

[15] Q. Vo and N. Foo. Reasoning about action: An argumentation-theoretic approach. *Journal of Artificial Intelligence Research*, 24:465–518, 2005.