Dialogues that Account for Different Perspectives in Collaborative Argumentation

Elizabeth Black
Department of Engineering Science
University of Oxford, UK
lizblack@robots.ox.ac.uk

Katie Atkinson
Department of Computer Science
University of Liverpool, UK
katie@liverpool.ac.uk

ABSTRACT

It is often the case that agents within a system have distinct types of knowledge. Furthermore, whilst common goals may be agreed upon, the particular representations of the individual agents' views of the world that they operate within may not always match. In this paper we provide a framework to allow different agents with different expertise to make individual contributions to an overall reasoning process, in order to make a decision about how to act to achieve some goal. Our framework is based on a model of argumentation that embeds inquiry dialogues within a process of practical reasoning. We combine two different approaches to argumentative reasoning and show not only how they can function together within a formal framework to provide richer interactions, but also how this facilitates reasoning across distributed agents who may each have different perspectives on the scenarios they operate in.

Categories and Subject Descriptors

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

Keywords

dialogue, argumentation, inquiry, persuasion, action

1. INTRODUCTION

In this paper we present a dialogue framework that allows agents with different spheres of expertise to inquire about the beliefs of others and to share specialist knowledge about the effect of actions. This allows them all to contribute to the decision of how to act to achieve some goal, despite their heterogeneous views of the world. We use an argumentation model that allows defeasible reasoning about what to believe, and a different argumentation model that allows defeasible reasoning about what to do (both defined in Section 2). In Section 3 of the paper we present the dialogue framework, which allows the agents not only to combine these two argumentation models but also allows them each to collaborate within the argumentation process. In Section 4 we demonstrate how this framework can be used, with an example based upon reasoning about the medical treatment of a patient. Section 5 concludes the paper.

The main contribution here is the first formal framework for multi-agent dialogues over actions which combines inquiry dialogue over beliefs with persuasion dialogue over actions. Although approaches to combining epistemic reasoning with practical reasoning exist [11, 12], these are focussed on argument-based semantics and do not provide a dialogue framework that allows multiple agents to collaboratively reason in this manner. Our framework guarantees that all knowledge relevant to both types of reasoning will be put forward during the dialogue. We do not require the agents to be aware of or agree upon all information relevant to the problem scenario: each agent has its own area of expertise and its own perspective on the problem. Allowing this distribution of knowledge significantly reduces the state space that each agent has to search when constructing arguments. Pooling all knowledge would be undesirable due to the increase in the state space. Also, there may be privacy issues and the communication costs of pooling the knowledge may be prohibitive.

Cite as: Dialogues that Account for Different Perspectives in Collaborative Argumentation, Elizabeth Black, Katie Atkinson, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 867–874

2. ARGUMENTATION MODELS USED

Agents within our framework may contain epistemic knowledge (*beliefs*) as well as normative knowledge about the effect of actions. We adapt García and Simari's Defeasible Logic Programming (DeLP) [7] for representing an agent's beliefs. DeLP is a formalism that combines logic programming with defeasible argumentation. It allows an agent to reason with inconsistent and incomplete knowledge that may change dynamically over time. Although we do not present it here, DeLP provides a dialectical reasoning mechanism for deciding whether an argument is acceptable. We assume that a proposition is a ground atom p and a literal is either a proposition p or a strongly negated proposition $\neg p$. Unlike in [7], we assume all knowledge to be defeasible.

Definition 1: *A* **defeasible rule** *is denoted* $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \alpha_0$ *where* α_i *is a literal for* $0 \leq i \leq n$. *A* **defeasible fact** *is denoted* α *where* α *is a literal. A* **belief** *is either a defeasible rule or a defeasible fact.* $\mathcal{B}$ *denotes the set of all beliefs.*

Each agent is identified by a unique id x taken from a set $\mathcal{I}$. Each agent has a, possibly inconsistent, belief base.

Definition 2: *A* **belief base** *associated with an agent* x *is a finite set of beliefs, denoted* Σ^x.

We now slightly adapt the definition of a defeasible derivation from [7] to deal with our assumption that all beliefs are defeasible.

Definition 3: *Let* Ψ *be a set of beliefs and* α *a literal. A* **defeasible derivation** *of* α *from* Ψ*, denoted* $\Psi \mid\!\sim \alpha$*, is a finite sequence* $\alpha_1, \alpha_2, \ldots, \alpha_n$ *of literals s.t.:* α_n *is* α*; and each literal* α_m *(*$1 \leq m \leq n$*) is in the sequence because either* α_m *is a defeasible fact in* Ψ*, or there exists a defeasible rule* $\beta_1 \wedge \ldots \wedge \beta_j \rightarrow \alpha_m$ *in* Ψ *s.t. every literal* β_i *(*$1 \leq i \leq j$*) is an element* α_k *preceding* α_m *in the sequence (*$k < m$*).*

We now define a *b-argument* as being a minimally consistent set of beliefs from which the claim can be defeasibly derived.

Definition 4: *A* **b-argument** *constructed from a set of, possibly inconsistent, beliefs Ψ is a tuple $\langle \Phi, \phi \rangle$ where ϕ is a defeasible fact and Φ is a set of beliefs s.t.: $\Phi \subseteq \Psi$; $\Phi \mid\sim \phi$; $\forall \phi, \phi'$ s.t. $\Phi \mid\sim \phi$ and $\Phi \mid\sim \phi'$, it is not the case that $\phi \cup \phi' \vdash \perp$ (where $\vdash$ represents classical implication); and there is no subset of Φ satisfying (1-3). Φ is called the* **support** *of the b-argument and ϕ is called the* **claim**.

We now describe the model of argumentation that we use to allow agents to reason about how to act. Although reasoning about what to do has been examined in numerous different contexts within the literature on multi-agent systems, it has recently received attention within argumentation-based accounts. In this paper we adapt one such account that is based upon the representation of an argumentation scheme and critical questions for practical reasoning as an action-based alternating transition system (AATS) [1]. Argumentation schemes are patterns of reasoning that, when instantiated, provide presumptive justification for the particular conclusion of the scheme [14]. Schemes are associated with a set of characteristic critical questions (CQs) that can be used to identify challenges to these justifications. The practical reasoning argument scheme given in [1], which we make use of in this paper, is as follows:

In the current circumstances R, we should perform action A, which will result in new circumstances S, which will realise goal G, which will promote some value V.

This scheme makes use of what are termed 'values' to describe some social interest that the agent wishes to uphold by realising the goal stated [3]. Values are used as a mechanism to represent the particular social interests of the agents in the system and they provide qualitative reasons as to why goals are desirable. So, an agent may propose an action, plus justification for its performance, by instantiating the scheme. However, the conclusion is a presumptive argument so an agent who does not accept this argument may challenge elements in the instantiation, through the application of CQs. An unfavourable answer to a CQ will identify a potential flaw in the argument. For example, one of the CQs associated with the scheme (CQ9) is 'Does doing the action have a side effect which demotes some other value?'. Through the CQs agents can attack the validity of the various elements of the argument scheme and the connections between them, suggest alternative possible actions, and point out side effects of the proposed action.

In order to be able to automate the reasoning embodied through the use of such a scheme, it needs to be grounded within some well-defined representation. In [1] such a formalism is presented to describe this scheme for practical reasoning in terms of an Action-based Alternating Transition System (AATS). AATSs are presented in [15] as structures for modelling game-like, dynamic, multi-agent systems in which the agents can perform actions in order to modify and attempt to control the system in some way. These structures serve as the basis for the representation of arguments about action in [1]. The formalism presented there provides a well-specified basis for addressing the problems of practical reasoning as presumptive argumentation in a multi-agent context, extending the formalism of [15] to enable the representation of values, where whether a value is promoted or demoted by a given action is determined by comparing the state reached with the state left.

Whilst the formalisms given in [1, 15] are intended to represent the overall behaviour of a multi-agent system and the effects of joint actions performed by the agents, we are interested in representing the specialist knowledge of the individual agents within a system. Hence, we adapt their formalisms to define our *Value-based Transition System* (VATS) as follows.

Definition 5: *A* **VATS** *for an agent x, denoted S^x, is a 9-tuple $\langle Q^x, q_0^x, Ac^x, Av^x, \rho^x, \tau^x, \Phi^x, \pi^x, \delta^x \rangle$ s.t.:*

Q is a finite set of states*;*

$q_0^x \in Q^x$ is the designated initial state*;*

Ac^x is a finite set of actions*;*

Av^x is a finite set of values*;*

$\rho^x : Ac^x \mapsto 2^{Q^x}$ is an action precondition function*, which for each action $a \in Ac^x$ defines the set of states $\rho(a)$ from which a may be executed;*

$\tau^x : Q^x \times Ac^x \mapsto Q^x$ is a partial system transition function*, which defines the state $\tau^x(q, a)$ that would result by the performance of a from state q—note that, as this function is partial, not all actions are possible in all states (cf. the precondition function above);*

Φ^x is a finite set of atomic propositions*;*

$\pi^x : Q^x \mapsto 2^{\Phi^x}$ is an interpretation function, which gives the set of primitive propositions satisfied in each state: if $p \in \pi^x(q)$, then this means that the propositional variable p is satisfied (equivalently, true) in state q; and

$\delta^x : Q^x \times Q^x \times Av^x \mapsto \{+, -, =\}$ is a valuation function *which defines the* status *(promoted (+), demoted (−), or neutral (=)) of a value $v \in Av^x$ ascribed by the agent to the transition between two states: $\delta^x(q, q', v)$ labels the transition between q and q' with respect to the value $v \in Av^x$.*

Note, $Q^x = \emptyset \leftrightarrow Ac^x = \emptyset \leftrightarrow Av^x = \emptyset \leftrightarrow \Phi^x = \emptyset$.

Given its VATS, an agent can now instantiate the practical reasoning argument scheme in order to construct arguments for and against actions—called *a-arguments*.

Definition 6: *An* **a-argument** *constructed by an agent x from its VATS S^x is a 6-tuple $A = \langle q_x, a, q_y, p, v, s \rangle$ s.t.: $q_x = q_0^x$; $a \in Ac^x$; $\tau^x(q_x, a) = q_y$; $p \in \pi^x(q_y)$; $v \in Av^x$; $\delta^x(q_x, q_y, v) = s$ where $s \in \{+, -, =\}$.*

We define the following functions: $\mathsf{Action}(A) = a$; $\mathsf{Goal}(A) = p$; $\mathsf{Value}(A) = v$; $\mathsf{Polarity}(A) = s$.

If $\mathsf{Polarity}(A) = +(-resp.)$*, then we say A is an a-argument for (against resp.) action a to achieve goal p.*

We now define arguments that instantiate CQs used to challenge a-arguments. For the sake of space, we consider only a subset of the CQs given in [1], being those we use in our example. We follow the numbering of CQs used in [1].

Definition 7: *A* **cq6-argument** *constructed from a VATS S^x is an a-argument $\langle q_x, a, q_y, p, v, + \rangle$ constructed from S^x. It challenges another a-argument $\langle q'_x, a', q'_y, p', v', + \rangle$ (answering the question* 'Are there alternative ways of realising the same goal?'*) iff $a \neq a'$ and $p = p'$.*

Definition 8: *A* **cq9-argument** *constructed from a VATS S^x is a 5-tuple $\langle q_x, a, q_y, v, - \rangle$ s.t.: $q_x = q_0^x$; $a \in Ac^x$; $\tau^x(q_x, a) = q_y$; $v \in Av^x$; $\delta^x(q_x, q_y, v) = -$. It challenges an a-argument $\langle q'_x, a', q'_y, p', v', + \rangle$ (answering the question* 'Does doing the action have a side effect which demotes some other value?'*) iff $a = a'$ and $v \neq v'$.*

Definition 9: *A* **cq10-argument** *constructed from a VATS S^x is a 5-tuple $\langle q_x, a, q_y, v, + \rangle$ s.t.: $q_x = q_0^x$; $a \in Ac^x$; $\tau^x(q_x, a) = q_y$; $v \in Av^x$; $\delta^x(q_x, q_y, v) = +$. It challenges an a-argument $\langle q'_x, a', q'_y, p', v', + \rangle$ (answering the question* 'Does doing the action have a side effect which promotes some other value?'*) iff $a = a'$ and $v \neq v'$.*

We refer to a-, cq6-, cq9- and cq10-arguments collectively as **act-arguments**. The set of all act-arguments that can be constructed by an agent x with VATS S^x is denoted $\mathcal{A}(S^x)$.

In [1], details are given of how the reasoning with the argument scheme and CQs is split into three stages : *problem formu-*

lation, where the agents decide upon the facts and values relevant to the particular situation under consideration; *epistemic reasoning*, where the agents determine the current situation with respect to the structure formed at the previous stage; and, *action selection* where the agents develop, and evaluate, arguments and counter arguments about what to do by instantiating the argument scheme and CQs. In [1] the authors demonstrate how agents, given their AATS representations of a given scenario, can produce arguments to reveal disagreements with other agents relevant to each of the three stages of reasoning described. However, what has not been considered previously is *how* within the epistemic reasoning stage the initial state can be determined in a situation of ignorance. In the approach that we describe here, we show how this can be determined through the use of an inquiry dialogue in which agents exchange arguments to establish what the state of the world is. The inquiry element thus allows for richer reasoning that considers agents' beliefs, in addition to their arguments about how to act.

3. THE DIALOGUE FRAMEWORK

Our framework assumes a closed cooperative multi-agent system. It allows an agent to take into account the various different views of the world held by the other specialist agents when making a decision about how to act in order to achieve a goal. In order to do this, the agents participate in a *persuasion over action* (pAct) dialogue in which each agent asserts all the act-arguments that it can construct which may be relevant to the decision. Once all such arguments have been presented, the agent who initiated the dialogue is able to use its own personal preference ordering over the values represented to decide which action to perform to achieve its goal.

In order to use its VATS to construct arguments, a participant x in a pAct dialogue must first establish what the current state (q_0^x) is, and so must determine the perceived truth value of each proposition p in Φ^x. It does this by constructing b-arguments for and against p and establishing whether such b-arguments are acceptable given the argumentation semantics it is using. However, in order to take full advantage of the various types of specialist knowledge available, the agent will construct such arguments and determine their acceptability status *collaboratively* with the other agents. They do this by entering into an *inquiry* sub-dialogue. This idea of shifting between different dialogue types has been explored in general terms (e.g. [9, 13]), here we give a specific protocol that allows shifts between pAct and inquiry dialogues.

We define our dialogue in the style of a dialogue game. Dialogue games are normally made up of a set of communicative acts called *moves*, a set of rules stating which moves it is legal to make at any point in a dialogue (the *protocol*), a set of rules defining the effect of making a move, and a set of rules that determine when a dialogue terminates (e.g. [9]). We use the general dialogue framework presented in [4], adapting it to allow for more than two participating agents. In [4], a protocol is given for inquiry dialogues that allow two agents to jointly construct b-arguments and determine their acceptability; here we provide a novel protocol for pAct dialogues that allow embedded inquiry sub-dialogues. As in [4], we also provide a *strategy*, which allows an agent to select exactly one legal move as the move to make.

We assume that there are always at least two agents (*participants*) taking part in a dialogue, each with its own identifier taken from the set $\mathcal{I}$. A move in our system is of the form $\langle Ag, Act, Det\rangle$. Ag is the identifier of the agent who makes the move (the *sender* of the move), Act is the type of move, and the Det gives the details of the move. The format for moves used in pAct dialogues is shown in Table 1, and the set of all moves meeting the format defined in Table 1 is denoted $\mathcal{M}$. Also, $\mathsf{Sender} : \mathcal{M} \mapsto \mathcal{I}$ is a function that returns the sender of a move.

Making the move $\langle x, open, dialogue(inq, \Delta, \Lambda)\rangle$ causes the participants to enter into an inquiry sub-dialogue. We do not give the details of an inquiry dialogue here but we assume that:

1. it fits with our upcoming definitions of the dialogue framework and allows multiple participants;
2. the topic is the set of propositions Δ;
3. the dialogue is guaranteed to terminate;
4. the dialogue outcome is a set of b-arguments with claim either p or $\neg p$ where $p \in \Delta$, such that a b-argument is part of the outcome of the dialogue if and only if it is an acceptable b-argument when reasoning with the union of the participating agents' beliefs. (Although we have presented a version of DeLP [7] as the argumentation model for epistemic reasoning, any suitable argumentation model could be substituted for this, allowing the use of different argumentation semantics.)

We are not aware of any existing dialogue systems that exactly fit our needs here. The one that most closely matches our requirements is given in [4], where a dialogue is defined for exactly two participants that will return as outcome a single acceptable argument for a proposition that is its topic; we believe it is possible to adapt this framework to meet our needs, as the difficult issues associated with multi-party dialogues (e.g. [5]) do not arise here due to the collaborative and exhaustive nature of the dialogue.

Once an inquiry dialogue has terminated, the agent x who initiated the dialogue must use the set of arguments for and against the propositions in Δ to decide whether $p \in \pi^x(q_0^x)$ for each $p \in \Delta$. We will assume here that the agent believes that $p \in \pi^x(q_0^x)$ if and only if there are n acceptable arguments for p and m acceptable arguments against p and $n > m$. Note however that this is a simplistic view and that, although it is beyond the scope of this paper, this problem of argument aggregation is an ongoing topic of interest within the community (e.g. [8, 10]).

As a dialogue progresses over time, we denote each timepoint by a natural number. A dialogue is simply a sequence of moves, each of which is made from one participant to all the other participants (i.e. at each timepoint, the agent whose turn it is sends the same move to every other participant of the dialogue). Agents may have more than one turn in a row. Each move is indexed by the timepoint when the move was made. The dialogue itself is indexed with two timepoints, indexing the first and last moves of the dialogue.

Definition 10: *A* **dialogue**, *denoted D_r^t, is a sequence of moves of the form $[m_r, \ldots, m_t]$ involving n participants $\{x_1, \ldots, x_n\}$ ($2 \leq n$) s.t. $\{x_1, \ldots, x_n\} \subseteq \mathcal{I}$; for all $1 \leq j, k \leq n$ s.t. $j \neq k$, $x_j \neq x_k$; $r, t \in \mathbb{N}$; and the following conditions hold:*

1. m_r is of the form $\langle x_i, open, dialogue(\theta, \gamma, [x_1, \ldots, x_n])\rangle$,

2. $\mathsf{Sender}(m_s) \in \{x_1, \ldots, x_n\}$ ($r \leq s \leq t$).

The **type** *of the dialogue D_r^t is given by $\mathsf{Type}(D_r^t) = \theta$. The* **topic** *of the dialogue D_r^t is given by $\mathsf{Topic}(D_r^t) = \gamma$. The* **initiator** *of the dialogue D_r^t is given by $\mathsf{Initiator}(D_r^t) = x_i$. The* **ordered participants** *of the dialogue D_r^t is given by $\mathsf{Participants}(D_r^t) = [x_1, \ldots, x_n]$. The set of all dialogues is denoted $\mathcal{D}$.*

The first move of a dialogue D_r^t must always be an open move (condition 1 of the previous definition) that gives a list of the participants which determines the order in which the agents take their turn, and every move of the dialogue must be made from a participant of the dialogue (condition 2).

We now define some terminology that allows us to talk about the relationship between two dialogues.

Definition 11: *Let D_r^t and $D_{r_1}^{t_1}$ be two dialogues. $D_{r_1}^{t_1}$ is a* **sub-dialogue** *of D_r^t iff $D_{r_1}^{t_1}$ is a sub-sequence of D_r^t ($r < r_1 \leq t_1 \leq t$).*

Move	Format
open	$\langle x, open, dialogue(\theta, \gamma, \Lambda)\rangle$
assert	$\langle x, assert, \Psi\rangle$
close	$\langle x, close, dialogue(\theta, \gamma, \Lambda)\rangle$

Table 1: The format for moves used in pAct dialogues, where either $\theta = pAct$ and γ is a proposition, or $\theta = inq$ and γ is a set of propositions; Λ is a list of agents ($\Lambda = [x_1, \ldots, x_n]$, $\{x_1, \ldots, n_n\} \subseteq \mathcal{I}$); Ψ is a set of act-arguments; and x is an agent ($x \in \mathcal{I}$).

D_r^t *is a* **top-level** *dialogue iff* $r = 1$. D_1^t *is a* **top-dialogue of** D_r^t *iff either the sequence* D_1^t *is the same as the sequence* D_r^t *or* D_r^t *is a sub-dialogue of* D_1^t. *If* D_r^t *is a sequence of* n *moves,* $D_r^{t_2}$ **extends** D_r^t *iff the first* n *moves of* $D_r^{t_2}$ *are the sequence* D_r^t.

In order to terminate a dialogue with n participants, n close moves must appear next to each other in the sequence (called a *matched-close*).

Definition 12: *Let* D_r^t *be a dialogue with participants* $\{x_1, \ldots, x_n\}$ *s.t.* $\mathsf{Type}(D_r^t) = \theta$, $\mathsf{Topic}(D_r^t) = \gamma$ *and* $\mathsf{Participants}(D_r^t) = \Lambda$. *We say that* m_s *(*$r < s \leq t$*) is a* **matched-close for** D_r^t *iff for all* i *s.t.* $0 \leq i \leq n$, $m_{s-i} = \langle \_, close, dialogue(\theta, \gamma, \Lambda)\rangle$.

So a matched-close will terminate a dialogue D_r^t but only if D_r^t has not already terminated and any sub-dialogues that are embedded within D_r^t have already terminated.

Definition 13: *Let* D_r^t *be a dialogue.* D_r^t **terminates at** t *iff the following conditions hold:*

1. m_t *is a matched-close for* D_r^t,
2. $\neg\exists D_r^{t_1}$ *s.t.* $D_r^{t_1}$ *terminates at* t_1 *and* D_r^t *extends* $D_r^{t_1}$,
3. $\forall D_{r_1}^{t_1}$ *if* $D_{r_1}^{t_1}$ *is a sub-dialogue of* D_r^t,
    then $\exists D_{r_1}^{t_2}$ *s.t.* $D_{r_1}^{t_2}$ *terminates at* t_2
        and either $D_{r_1}^{t_2}$ *extends* $D_{r_1}^{t_1}$ *or* $D_{r_1}^{t_1}$ *extends* $D_{r_1}^{t_2}$,
        and $D_{r_1}^{t_2}$ *is a sub-dialogue of* D_r^t,
        and $(t - t_2) \leq \mid \mathsf{Participants}(D_r^t) \mid$.

As it is possible to have multiple nested dialogues, it is sometimes useful to refer to the current dialogue, which is the innermost dialogue that has not yet terminated.

Definition 14: *Let* D_r^t *be a dialogue. The* **current dialogue** *is returned by* $\mathsf{Current}(D_r^t)$ *s.t.* $\mathsf{Current}(D_r^t) = D_{r_1}^t$ *(*$1 \leq r \leq r_1 \leq t$*) where the following conditions hold:*

1. $m_{r_1} = \langle x, open, \gamma\rangle$ *for some* $x \in \mathcal{I}$ *and some* $\gamma \in \mathcal{B}$,
2. $\forall D_{r_2}^{t_1}$ *if* $D_{r_2}^{t_1}$ *is a sub-dialogue of* $D_{r_1}^t$,
    then $\exists D_{r_2}^{t_2}$ *s.t. either* $D_{r_2}^{t_2}$ *extends* $D_{r_2}^{t_1}$ *or*
            $D_{r_2}^{t_1}$ *extends* $D_{r_2}^{t_2}$,
        and $D_{r_2}^{t_2}$ *is a sub-dialogue of* $D_{r_1}^t$
        and $D_{r_2}^{t_2}$ *terminates at* t_2,
3. $\neg\exists D_{r_1}^{t_3}$ *s.t.* $D_{r_1}^t$ *extends* $D_{r_1}^{t_3}$ *and* $D_{r_1}^{t_3}$ *terminates at* t_3.

The order in which the participants take their turn is determined by the order the participants appear in the list given in the move opening the dialogue. The only exception to this is if the last move to be made was a matched-close terminating a sub-dialogue, in which case it is the turn of the agent who opened said sub-dialogue.

Definition 15: *Let* D_r^t *be a dialogue that has not terminated s.t.* $\mathsf{Current}(D_r^t) = D_{r_1}^t$ *and* $\mathsf{Participants}(D_{r_1}^t) = [x_1, \ldots, x_n]$. *The agent whose* **turn** *it is next to move is given by* $\mathsf{Turn}(D_r^t)$ *s.t.:*

if $D_{r_2}^t$ *is a sub-dialogue of* D_r^t *that terminates at* t, *then* $\mathsf{Turn}(D_r^t) = \mathsf{Initiator}(D_{r_2}^t)$;

else if $\mathsf{Sender}(m_t) = x_n$, *then* $\mathsf{Turn}(D_r^t) = x_1$;

else if $\mathsf{Sender}(m_t) = x_i$ *s.t.* $1 \leq i < n$, *then* $\mathsf{Turn}(D_r^t) = x_{i+1}$.

We adopt the standard approach of associating a *commitment store* with each agent participating in a dialogue. A commitment store is a set of everything that the agent has asserted so far in the course of the dialogue. Agents assert act-arguments during pAct dialogues and we assume that they only assert b-arguments during inquiry dialogues (although this will depend on the exact details of the dialogue framework being used). As a commitment store consists of things that the agent has already publicly declared, its contents are visible to the other agents participating in the dialogue.

Definition 16: *A* **commitment store** *associated with an agent* x *at a timepoint* t, *denoted* CS_x^t, *where* $x \in \mathcal{I}$ *and* $t \in \mathbb{N}$, *is a set of act-arguments and b-arguments.*

An agent's commitment store grows monotonically over time. If an agent makes a move asserting a set of act-arguments, they are all added to the agent's commitment store. This is the only time the commitment store is updated during a pAct dialogue.

Definition 17: **Commitment store update.** *For a pAct dialogue with participants* $\{x_1, \ldots, x_n\}$, *for all* $x \in \{x_1, \ldots, x_n\}$,

$$CS_x^t = \begin{cases} \emptyset & \textit{iff } t = 0, \\ CS_x^{t-1} \cup \Psi & \textit{iff } m_t = \langle x, assert, \Psi\rangle, \\ CS_x^{t-1} & \textit{otherwise.} \end{cases}$$

A protocol is a function that returns the set of moves that are legal for an agent to make at a particular point in a particular type of dialogue. Here we give the specific protocol for pAct dialogues. It takes the top-level dialogue that the agents are participating in and the identifier of the agent whose turn it is to move, and returns the set of legal moves that the agent may make.

Definition 18: *The* **pAct protocol** *is a function* $\Pi : \mathcal{D} \times \mathcal{I} \mapsto \wp(\mathcal{M})$. *If* D_1^t *is a top-level dialogue s.t.* $\mathsf{Current}(D_1^t) = D_r^t$, $\mathsf{Turn}(D_r^t) = x$, $\mathsf{Participants}(D_r^t) = \Lambda = [x_1, \ldots, x_n]$, $CSs = \bigcup_{\forall x_i \in \{x_1, \ldots, x_n\}} CS_{x_i}^t$, $\mathsf{Type}(D_r^t) = pAct$, $\mathsf{Topic}(D_r^t) = p$ *and* $1 \leq t$, *then* $\Pi(D_1^t, x)$ *is*

$$\Pi_{\mathsf{o}}(D_1^t, x) \cup \Pi_{\mathsf{a}}(D_1^t, x) \cup \{\langle x, close, dialogue(pAct, p, \Pi)\rangle\}$$

where

$\Pi_{\mathsf{o}}(D_1^t, x) = \{\langle x, open, dialogue(inq, \Delta, \Lambda)\rangle \mid$
    $\neg\exists t'$ *s.t.* $1 < t' \leq t$
        and $m_{t'} = \langle x, open, dialogue(inq, \Delta, \Lambda)\rangle\}$

$\Pi_{\mathsf{a}}(D_1^t, x) = \{\langle x, assert, \Psi\rangle \mid$
    (1) $\Psi \neq \emptyset$, *and*
    (2) $\forall A \in \Psi$:
        (i) $A \notin CSs$, *and*
        either (ii,a) $\mathsf{Goal}(A) = p$, $\mathsf{Action}(A) = a$,
            and $\mathsf{Polarity}(A) = +$ *and* $\neg\exists A' \in CSs$ *s.t.*
            $\mathsf{Action}(A') = a$
        or (ii,b) $A = \langle q_x, a, q_y, v, -\rangle$ *and* $\exists A' \in CSs$ *s.t.*
            $\mathsf{Action}(A') = a$, $\mathsf{Value}(A) = v' (v' \neq v)$
            and $\mathsf{Polarity}(A') = +$,
        or (ii,c) $A = \langle q_x, a, q_y, v, +\rangle$ *and* $\exists A' \in CSs$ *s.t.*
            $\mathsf{Action}(A') = a$, $\mathsf{Value}(A) = v'$ *(*$v' \neq v$*)*
            and $\mathsf{Polarity}(A) = +$.

else $\Pi(D_1^t, x) = \emptyset$.

The definition of $\Pi_{\mathsf{o}}(D_1^t, x)$ ensures the participants can enter into inquiry sub-dialogues to jointly construct all of the acceptable b-arguments for and against each proposition $p \in \Delta$, allowing the initiating agent to decide what it believes the truth value of p to be.

The definition of $\Pi_{\mathsf{a}}(D_1^t, x)$ ensures that an agent can assert any act-argument relevant to the initiator's proposal about how to act. An agent may not assert the empty set (condition 1) nor assert arguments that have already been asserted (2i). An agent may assert: any a-argument for an action to achieve the goal (the topic of the dialogue) as long as no a-argument for the same action has already been asserted (2iia) (allowing arguments to be put forward when the commitment stores are empty and allowing any cq6-argument for an action that has not yet been considered to be put forward); any cq9-argument that challenges an a-argument that has already been asserted for an action to achieve the goal (2iib); any cq10-argument that challenges an a-argument that has already been asserted for an action to achieve the goal (2iic). We have only used three CQs here, but could apply our approach to deal with the others. Note, it is straightforward to check conformance with the protocol as it only refers to public elements of the dialogue.

We now give a specific strategy that allows an agent to select exactly one legal move to make at each timepoint (at which it is its turn) in a pAct dialogue. A strategy is personal to an agent (i.e. the move it returns depends on the agent's private beliefs). The strategy states that if it is legal to make an open move that opens an inquiry sub-dialogue whose topic is Φ^x (recall Φ^x is the set of propositions that an agent x uses to model the world), then make any such a move; else, if x can construct a set of act-arguments from its VATS such that asserting them is a legal move, then assert the maximal (with regards to set inclusion) such set; else make a close move.

Definition 19: *The* **pAct strategy** *is a function* $\Omega : \mathcal{D} \times \mathcal{I} \mapsto \mathcal{M}$ *s.t.* $\Omega(D_1^t, x)$ *is defined as follows (where* $\mathsf{Turn}(D_1^t) = x$*).*

If $\langle x, open, dialogue(inq, \Phi^x, \Lambda)\rangle \in \Pi(D_1^t, x)$,
then $\Omega(D_1^t, x) = \langle x, open, dialogue(inq, \Phi^x, \Lambda)\rangle$
else,
if $\exists \Psi \subseteq \mathcal{A}(S^x)$ *s.t.* $\langle x, assert, \Psi\rangle \in \Pi(D_1^t, x)$
and $\neg\exists \Psi' \subset \Psi$ *s.t.* $\langle x, assert, \Psi\rangle \in \Pi(D_1^t, x)$,
then $\Omega(D_1^t, x) = \langle x, assert, \Psi\rangle$;
else,
if $\exists\langle x, close, dialogue(pAct, p, \Lambda)\rangle \in \Pi(D_1^t, x)$,
then $\Omega(D_1^t, x) = \langle x, close, dialogue(pAct, p, \Lambda)\rangle$.

The strategy ensures that before asserting any act-arguments, an agent will have entered into an inquiry sub-dialogue for each of the propositions represented in its model of the world, meaning it will be in a position to take into account all of the heterogeneous knowledge held by the different specialist agents within the system when deciding what its initial state is. After an agent has established its initial state, it will assert the set of all act-arguments that it can construct which are legal to assert. An agent only makes a close move if it cannot make an open or assert move and so, as a dialogue only terminates when each participating agent has made a close move, the dialogue will not terminate until every relevant act-argument that the agents can construct has been asserted.

3.1 Properties of the Framework

We can show that all pAct dialogues where the participants are each following the pAct strategy terminate (as assumed earlier, inquiry sub-dialogues terminate; each agent x will only make an open move opening an inquiry sub-dialogue with Φ^x as its topic and agents cannot repeat these moves; and there are a finite number of act-arguments that can be generated from the agents' VATSs and these cannot be asserted more than once).

Proposition 1: *Let* D_r^t *be a pAct dialogue in which the participants are following the pAct strategy. There exists a* t_1 *(*$r < t \leq t_1$*) such that* $D_r^{t_1}$ *terminates at* t_1 *and* $D_r^{t_1}$ *extends* D_r^t*.*

We are also able to show that for a terminated pAct dialogue with topic p and participants that are each following the pAct strategy: any act-argument that is asserted during the dialogue can be constructed by at least one of the agents; if any of the participants can construct an a-argument for an action to achieve the goal p, then an a-argument for that action will have been asserted during the dialogue (covering CQ6 'Are the alternative ways to achieve the goal?'); and, for any a-argument A that gets asserted during the dialogue, if any participant can construct a cq9-, or cq10-argument A' that challenges the a-argument A, then A' will have been asserted during the dialogue. This follows from the definitions of the pAct protocol and the pAct strategy, and the assumptions that we make about the inquiry sub-dialogue.

Proposition 2: *Let* D_r^t *be a pAct dialogue with participants* $Ag = \{x_1, \ldots, x_n\}$ *who each follow the pAct strategy s.t.* D_r^t *terminates at t and* $\mathsf{Topic}(D_r^t) = p$*. Let* $CSs = \bigcup_{\forall x_i \in Ag} CS_{x_i}^t$ *and* $Args = \bigcup_{\forall x_i \in Ag} \mathcal{A}(S^{x_i})$*. Let us assume that as a result of reasoning with* $\bigcup_{\forall x_i \in Ag} \Sigma^{x_i}$*, each participant* $x_i \in Ag$ *determines its initial state to be* $q_0^{x_i}$*. The following conditions hold:*

1. if there exists an act-argument $A \in CSs$*, then* $A \in Args$*;*

2. there exists an a-argument $A \in CSs$ *for an action* a *to achieve goal* p *iff there exists an a-argument* $A' \in Args$ *for action* a *to achieve goal* p*;*

3. for any a-argument $A \in CSs$ *and any cq9- or cq10-argument* A' *that challenges A,* $A' \in CSs$ *iff* $A' \in Args$*.*

Our framework thus ensures that the initiator of a pAct dialogue will, at the end of the dialogue, have at hand all arguments that any participating agent can construct and that may be relevant to its decision as to how to act. (Recall, we only consider CQs 6, 9 and 10 here, but this approach is applicable to all CQs.) In the next section we give an example of a dialogue generated by our framework and an example of a mechanism the initiator may use to evaluate the relevant arguments produced from a pAct dialogue.

4. DIALOGUE EXAMPLE

Our example scenario concerns a system for reasoning about the medical treatment of a patient (this scenario is for illustrative purposes only, we do not make any claims about the validity of the medical knowledge). It is adapted from [2], where the agents' knowledge is only given informally and the mechanism the agents use to share their knowledge is not defined. Note, following [7], we use *schematic rules* in the example that contain variables (each schematic rule stands for all ground instances of the rule). We assume agents have knowledge of the basic mathematical operators.

4.1 Example Setting

In the scenario we consider three different specialist agents, each with a single unique value of concern: the Treatment Agent (TA), the Cost Agent (CA) and the Efficacy Agent (EA). The question to be answered in the scenario concerns the particular drug that should be prescribed to the patient to prevent blood clotting: aspirin, chlopidogrel or streptokinase. Prescribing these drugs gives the basis for three actions that are recognised by all agents:

$act1 : asp \qquad act2 : chlop \qquad act3 : strep$

We now present each of the agents in turn by considering subsets of the VATS that each maintains. We begin with the TA. This agent's sole concern, and hence its value, is the healthy recovery of the patient. Reasoning about the effects of the different drugs requires consideration of the following propositions:

p_1^t: *clotPrev*; whether the patient's blood has been prevented from clotting

p_2^t: *aspCont*; whether aspirin is contraindicated for the patient[1]

p_3^t: *gastRisk*; whether the patient is at risk of gastric ulceration

Actions cause transitions between states in which the truth values of the propositions may change (where 1 denotes *true* and 0 denotes *false*). The transitions and their effects relevant to this example

---

[1] To keep the example small we consider this only in respect of aspirin but the other drugs could be treated in a similar manner.

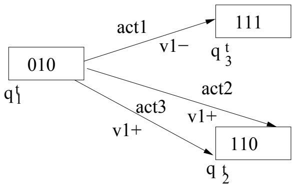

Figure 1: VATS for TA, S^{ta}

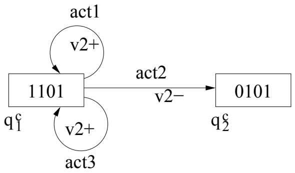

Figure 2: VATS for CA, S^{ca}

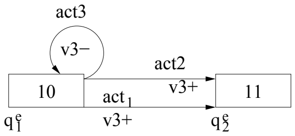

Figure 3: VATS for EA, S^{ea}

are given in Fig. 1. The transitions also show when the value of concern ($v1$) is promoted (+) or demoted (–).

As well as the information represented in its VATS about the actions that can be performed, the TA also has some knowledge about the domain, in the form of defeasible rules, as follows ($histGast$ indicates the patient has a history of gastritis and $acidRedTher$ indicates the patient is receiving acid reducing therapy).

$\Sigma^{ta} = \{histGast \wedge \neg acidRedTher \rightarrow aspContra\}$

The VATS for the CA (the relevant part of which is shown in Fig. 2) is different to that of the TA, reflecting its different area of expertise, and it shows only propositions relevant for its reasoning and how actions affect these in respect to its value 'money' ($v2$). The propositions of concern to this agent are:

p^c_1: *onBudg*; whether the budget for the patient is adhered to

p^c_2: *aspEcon*; whether prescribing aspirin to the patient keeps within budget

p^c_3: *chlopEcon*; whether prescribing chlopidogrel to the patient keeps within budget

p^c_4: *strepEcon*; whether prescribing streptokinase to the patient keeps within budget

As with the TA, the CA also contains some domain knowledge (where $spentBudg(X)$ indicates that amount of the budget that has already been spent on the patient is X).

$$\Sigma^{ca} = \{spentBudg(X) \wedge < (X, 750) \rightarrow onBudg,$$
$$spentBudg(X) \wedge < (1, 750 - X) \rightarrow aspEcon,$$
$$spentBudg(X) \wedge < (75, 750 - X) \rightarrow chlopEcon,$$
$$spentBudg(X) \wedge < (5, 750 - X) \rightarrow strepEcon\}$$

Finally, turning to the EA, whose concern is the treatment's effectiveness ($v3$), the following propositions are relevant to this agent:

p^e_1: *over50*; whether the patient is older than 50

p^e_2: *clotPrev*; effect the drug has on prevention of blood clotting

From the transitions shown in Fig. 3 we can see that prescribing streptokinase is the only action whose effectiveness is reduced for a patient over the age of 50.

The EA also contains some domain knowledge (where $age(X)$ indicates that the patient's age is X).

$\Sigma^{ea} = \{age(X) \wedge > (X, 50) \rightarrow over50\}$

As well as the three agents we have described above, there is a fourth agent who is the keeper of specific information relating to the patient in question—the Patient Agent (PA). The PA is unique to an individual patient, and can be seen as subsuming the role of electronic patient record. It is the PA who has the initial goal that triggers the dialogue: to prevent blood clotting (i.e. in the current state *clotPrev* is false, the PA's goal is to bring about a state in which *clotPrev* is true). The PA does not have any knowledge about the effect of actions and so initiates a pAct dialogue with the other specialist agents in order to collect arguments about what action could be taken in order to bring about the goal. Once the PA has elicited all of the relevant arguments, it is able to use its personal preference ordering over the values represented to decide which action to take. The information held by the PA that is relevant to this example is now given.

$\Sigma^{pa} = \{histGast, \neg acidRedTher, spentBudg(695), age(65)\}$

In the next section we give the details of the dialogue generated.

4.2 Dialogue Example

We first give the act-arguments that are generated by the agents from their VATSs during the dialogue.

A1:*Given that we are in state q^t_1, we should not give aspirin, which will prevent the patient's blood clotting and put the patient at risk of gastric ulceration, and so demotes the patient's healthy recovery (cq9-argument, challenges A7).*

$\langle q^t_1, asp, q^t_3, v1, - \rangle$

A2:*Given that we are in state q^t_1, we should give chlopidogrel, which will prevent the patient's blood clotting, and so promotes the patient's healthy recovery (a-argument to achieve topic of dialogue).*

$\langle q^t_1, chlop, q^t_2, clotPrev, v1, + \rangle$

A3:*Given that we are in state q^t_1, we should give streptokinase, which will prevent the patient's blood clotting, and so promotes the patient's healthy recovery (a-argument to achieve topic of dialogue).*

$\langle q^t_1, strep, q^t_2, clotPrev, v1, + \rangle$

A4:*Given that we are in state q^c_1, we should give aspirin, which will ensure we stay within budget for the patient, and so promotes the hospital's money (cq10-argument, challenges A7).*

$\langle q^c_1, asp, q^c_1, v2, + \rangle$

A5:*Given that we are in state q^c_1, we should not give chlopidogrel, which will put us out of the budget for the patient, and so demotes the hospital's money (cq9-argument, challenges A2).*

$\langle q^c_1, chlop, q^c_2, v2, - \rangle$

A6: *Given that we are in state q^c_1, we should give streptokinase, which will ensure we stay within budget for the patient, and so promotes the hospital's money (cq10-argument, challenges A3).*

$\langle q^c_1, strep, q^c_1, v2, + \rangle$

A7:*Given that we are in state q^e_1, we should give aspirin, which will prevent the patient's blood clotting, and so promotes effectiveness (cq6-argument, challenges A2 and A3).*

$\langle q^e_1, asp, q^e_2, clotPrev, v3, + \rangle$

A8:*Given that we are in state q^e_1, we should give chlopidogrel, which will prevent the patient's blood clotting, and so promotes effectiveness (cq10-argument, challenges A2).*

$\langle q^e_1, chlop, q^e_2, v3, + \rangle$

A9:*Given that we are in state q^e_1, we should not give streptokinase, which will not prevent the patient's blood clotting, and so demotes effectiveness (cq9-argument, challenges A3).*

$\langle q^e_1, strep, q^e_1, v3, - \rangle$

We now give the b-arguments that it is possible to construct from the union of all the participating agents' beliefs.

A10 $= \langle\{histGast, \neg acidRedTher,$
$histGast \wedge \neg acidRedTher \rightarrow aspContr\}, aspContr\rangle$,
A11 $= \langle\{spentBudg(695), < (695, 750), spentBudg(695) \wedge$
$< (695, 750) \rightarrow onBudg\}, onBudg\rangle$,
A12 $= \langle\{spentBudg(695), < (1, 55), spentBudg(695) \wedge$
$< (1, 55) \rightarrow aspEcon\}, aspEcon\rangle$,
A13 $= \langle\{spentBudg(695), < (5, 55), spentBudg(695) \wedge$
$< (5, 75) \rightarrow strepEcon\}, strepEcon\rangle$,
A14 $= \langle\{age(65), > (65, 50), age(65) \wedge$
$> (65, 50) \rightarrow over50\}, over50\rangle$

Note that as none of these arguments are in conflict with one another (i.e. there is no argument whose claim disputes the claim or premise of another argument), each of these arguments is acceptable, given that we are considering the union of the participating agents' beliefs. Hence, as a result of the inquiry sub-dialogues in which these b-arguments are elicited, the TA, CA and EA find themselves in initial states q_1^t, q_1^c and q_1^e respectively.

The dialogue generated by the framework is given in Table 2, where the first column gives the timepoint t, the second column gives the move m_t and the third shows how the various arguments get added to the commitment stores over the course of the dialogue.

4.3 Argument Evaluation

At this stage the PA needs some way to evaluate the arguments generated to decide which action to perform and why. Although our framework does not prescribe which evaluation mechanism should be used, we give an example that uses an abstract Argumentation Framework (AF) [6] extended to represent values, a Value-Based Argumentation Framework (VAF) [3]. An AF comprises a finite set of arguments and a binary attack relation between pairs of arguments. AFs can be modelled as directed graphs with arguments as nodes and attacks as edges. A maximal set of arguments which do not attack one another, but which between them attack every attacker of a member of the set is a *preferred extension* (PE) and represents a maximal consistent position. As described in [3], VAFs extend AFs by associating arguments with values that are promoted through acceptance of the argument, recognising the agents' different interests. In a VAF attacks succeed only if the value associated with the attacking argument is ranked, by the audience evaluating the VAF, equal to or higher than the argument attacked, unlike in AFs where attacks always succeed.

We now consider the VAF that the PA can construct and suppose that the agent's value order is Health (H) $>$ Efficiency (E) $>$ Money (M) ($v1 > v3 > v2$). Although the VAF comprises all the arguments generated in the example, those generated through the use of CQ10 are treated differently to those generated from instantiating the other CQs. CQ10 does not dispute the action that should be performed, but questions the justification for the conclusion i.e. why the action should be performed given the value it promotes. So when CQ10 is posed against an argument instantiating the argument scheme, each of the two *justifications* for the action attack the alternative. This is characterised as an attack since in some scenarios (e.g. inquiry into negligence) it might be that the conclusion is only supported by one of the justifications and not both, depending on the value ordering of the agent evaluating the VAF. However, both arguments generated still endorse the same conclusion, so we need some way of recognising and representing this. We could look to use a form of argument aggregation e.g. [8, 10], but there are numerous subtleties involved in argument aggregation that have yet to be resolved (for example, is a collection of very weak arguments stronger than one very strong argument?). Instead, we use the following mechanism.

Where two (or more) arguments providing different justifications for the same conclusion exist in the PE, as generated by CQ10, the arguments can be combined into a single argument, justified by both the values they each promote. Where these values are distinct, the value that is ranked most highly in the audience's preference ordering is the one that justifies the argument for that audience.

Applying the above procedure to our example means that A2 and A8 can be combined. So, for the audience H $>$ E $>$ M (as in our example), val(A2/8) = H. For a different audience, say with ranking E $>$ H $>$ M, val(A2/8) = E. Since CQ10 was also used to generate arguments A4 and A6, these can be treated analogously.

The full VAF for the arguments generated in the debate is given below in Fig. 4, where each argument is labelled with the value(s) that it either promotes or demotes.

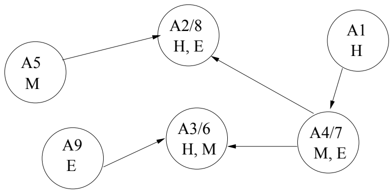

Figure 4: VAF for the arguments in the example.

To decide which action to take, the VAF is evaluated in relation to the PA's value ordering, H $>$ E $>$ M. So, starting with the arguments with no attackers we add A1, A5 and A9 to the PE. A1 defeats A4/7 given the value ordering so A2/8 and A3/6 are no longer attacked by A4/7. A5 attacks but does not defeat A2/8 due to the value ordering. Similarly, A9 attacks but does not defeat A3/6[2]. So, the PE for the VAF is the set {A1, A2/8, A3/6, A5, A9} from which we need to decide upon a final argument for action. First we consider the cq9-arguments that are in the PE and which conclude that an action should not be performed, such that this action is not endorsed by any other argument in the PE. The only such argument is A1, which proposes that aspirin should not be prescribed, and since no other argument in the PE proposes that it should be, A1 can be discounted.

We are now left with one argument each for $chlop$ (A2/8) and $strep$ (A3/6). There are many different rationales for deciding between such acceptable arguments, and the one chosen may be application dependant. For example, in safety-critical domains such as this, it may be desirable to have a human user make this final decision. Here, however, we consider the strength of the arguments in the PE that challenge either A2/8 or A3/6: A5 and A9. We can see that A5's value is M, the lowest ranked value, and A9's value is E. So, an audience with the value ordering given can consider A5 to have the least strength, allowing A2/8's conclusion, to prescribe chlopidogrel, to be the least strongly challenged and return this action as the one to execute. Alternatively, the PA could make further consultations with other agents to check if there are any remedies available to counter the demotion of values, as pointed out in A9 through the use of CQ9. We leave this example here.

5. CONCLUSION

In this paper we have presented the first formal dialogue framework to allow agents to combine an inquiry dialogue over beliefs with a persuasion dialogue over action. This allows agents with heterogeneous knowledge to each have an input into a decision about

---

[2] A9 would defeat A3/6 if val(A3/6) = M, which is not the case here given the value ordering we have assumed for the PA.

t	m_t	CSs
1	$\langle pa, open, dialogue(pAct, clotPrev, \Lambda)\rangle$	
2	$\langle ta, open, dialogue(inq, \{clotPrev, aspCont, gastRisk\}, \Lambda)\rangle$	
$\vdots$	$\vdots$	$A10$
$2 + n_1$	$\langle \_, close, dialogue(inq, \{clotPrev, aspCont, gastRisk\}, \Lambda)\rangle$	
$3 + n_1$	$\langle ta, assert, \{A2, A3\}\rangle$	$A2, A3$
$4 + n_1$	$\langle ca, open, dialogue(inq, \{onBudg, aspEcon, chlopEcon, strepEcon\}, \Lambda)\rangle$	
$\vdots$	$\vdots$	$A11, A12, A13$
$4 + n_1 + n_2$	$\langle \_, close, dialogue(inq, \{onBudg, aspEcon, chlopEcon, strepEcon\}, \Lambda)\rangle$	
$5 + n_1 + n_2$	$\langle ca, assert, \{A5, A6\}\rangle$	$A5, A6$
$6 + n_1 + n_2$	$\langle ea, open, dialogue(inq, \{over50, clotPrev\}, \Lambda)\rangle$	
$\vdots$	$\vdots$	$A14$
$6 + n_1 + n_2 + n_3$	$\langle \_, close, dialogue(inq, \{over50, clotPrev\}, \Lambda)\rangle$	
$7 + n_1 + n_2 + n_3$	$\langle ea, assert, \{A7, A8, A9\}\rangle$	$A7, A8, A9$
$8 + n_1 + n_2 + n_3$	$\langle pa, close, dialogue(pAct, clotPrev, \Lambda)\rangle$	
$9 + n_1 + n_2 + n_3$	$\langle ta, assert, \{A1\}\rangle$	$A1$
$10 + n_1 + n_2 + n_3$	$\langle ca, assert, \{A4\}\rangle$	$A4$
$11 + n_1 + n_2 + n_3$	$\langle ea, close, dialogue(pAct, clotPrev, \Lambda)\rangle$	
$12 + n_1 + n_2 + n_3$	$\langle pa, close, dialogue(pAct, clotPrev, \Lambda)\rangle$	
$13 + n_1 + n_2 + n_3$	$\langle ta, close, dialogue(pAct, clotPrev, \Lambda)\rangle$	
$14 + n_1 + n_2 + n_3$	$\langle ca, close, dialogue(pAct, clotPrev, \Lambda)\rangle$	

Table 2: pAct dialogue example: the PA is looking for the other specialist agents within the system (TA, CA, EA) to provide arguments relative to its decision about how to act to achieve goal $clotPrevent$. Note, $\Lambda = [pa, ta, ca, ea]$ and $CSs = \bigcup_{x \in \{pa,ta,ca,ea\}} CS^t_x$.

how to act to achieve a goal. The key features of our approach are: the dialogue framework caters for both arguments about belief as well as arguments about what to do; we have shown that the framework guarantees that all knowledge potentially relevant to both types of reasoning will be elicited in the course of the dialogue; our framework allows different areas of knowledge to be specialised to particular agents that may then each lend their own perspective on the problem; and, the distributed nature of the specialist knowledge significantly reduces each agents' state space representation of the scenario.

There are numerous avenues of future work for investigation. In particular, we would like to formalise: the evaluation stage that comes after an inquiry dialogue, where a decision must be made as to which of a set of propositions to believe given a set of conflicting arguments for and against those propositions; and, the evaluation stage that comes after a pAct dialogue, where a decision must be made as to which of a set of actions to carry out given a set of conflicting arguments for and against those actions.

6. REFERENCES

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

[2] K. Atkinson, T. J. M. Bench-Capon, and S. Modgil. Argumentation for decision support. In *17th Int. Conf. on Database and Expert Systems Applications*, LNCS 4080, pages 822–831. Springer, 2006.

[3] T. J. M. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *J. of Logic and Computation*, 13(3):429–48, 2003.

[4] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, DOI 10.1007/s10458-008-9074-5, 2009.

[5] F. Dignum and G. Vreeswijk. Towards a testbed for multi-party dialogues. In *AAMAS Int. Workshop on Agent Communication Languages and Conversation Policies*, pages 63–71, 2003.

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

[7] A. J. García and G. R. Simari. Defeasible logic programming an argumentative approach. *Theory and Practice of Logic Prog.*, 4(1–2):95–138, 2004.

[8] P. Krause, S. Ambler, M. Elvang-Goransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11(1):113–131, 1995.

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

[10] H. Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *10th Int. Conf. on Artificial Intelligence and Law*, pages 85–94. ACM Press, 2005.

[11] H. Prakken. Combining sceptical epistemic reasoning with credulous practical reasoning. In *Computational Models of Argument, Proc. of COMMA-06*, pages 311–322, 2006.

[12] I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. In *5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 347–354. ACM Press, 2006.

[13] C. Reed. Dialogue frames in agent communications. In *3rd Int. Conf. on Multi-Agent Systems*, pages 246–253, 1998.

[14] D. N. Walton. *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1996.

[15] M. Wooldridge and W. van der Hoek. On obligations and normative ability: Towards a logical analysis of the social contract. *J. of Applied Logic*, 3:396–420, 2005.