A Generative Inquiry Dialogue System

Elizabeth Black*  Anxiety Hunter
Department of Engineering Science Department of Computer Science
University of Oxford University College London
Oxford, UK London, UK
lizblack1@gmail.com A.Hunter@cs.ucl.ac.uk

ABSTRACT
The majority of existing work on agent dialogues considers negotiation, persuasion or deliberation dialogues. We focus on inquiry dialogues that allow two agents to share knowledge in order to construct an argument for a specific claim. Inquiry dialogues are particularly useful in cooperative domains such as healthcare, and can be embedded within other dialogue types. Existing inquiry dialogue systems only model dialogues, meaning they provide a protocol which dictates what the possible legal next moves are but not which of these moves to make. Our system not only includes a general dialogue-game style inquiry protocol but also a strategy, for an agent to use with this protocol, that selects exactly one of the legal moves to make. We propose a benchmark against which we compare our dialogues, being the arguments that can be constructed from the union of the agents’ beliefs, and use this to define soundness and completeness properties for inquiry dialogues. We show that these properties hold for all well-formed inquiry dialogues in our system.

Keywords

1. INTRODUCTION
Dialogue games are now a common approach to defining communicative agent behaviour, especially when this behaviour is argumentation-based (e.g. [9, 11]). Dialogue games are normally made up of a set of communicative acts called moves, a set of rules that state which moves it is legal to make at any point in a dialogue (the protocol), a set of rules that define the effect of making a move, and a set of rules that determine when a dialogue terminates. Most of the work so far has looked only at modelling different types of dialogue from the Walton and Krabbe typology [14], here we provide a generative system.

In this paper we focus on inquiry dialogues. Walton and Krabbe define an inquiry dialogue as arising from an initial situation of “general ignorance” and as having the main goal to achieve the “growth of knowledge and agreement”. Each individual participating in an inquiry dialogue has the goal to “find a ‘proof’ or destroy one” [14, page 66]. No formal definition of inquiry dialogues is given, leaving this classification somewhat open to interpretation. To address this, we have defined two different types of inquiry dialogue, each of which we believe fits this general definition: warrants and argument inquiry. In a warrant inquiry dialogue, the ‘proof’ takes the form of a dialectical tree (essentially a tree with an argument at each node, whose arcs represent the counter-argument relation and that has at its root an argument whose claim is the topic of the dialogue). In an argument inquiry dialogue, the ‘proof’ takes the form of an argument for the topic of the dialogue. Argument inquiry dialogues are commonly embedded in warrant inquiry dialogues. In this paper, we will focus only on argument inquiry dialogues.

As far as we are aware, there are only two groups that have proposed inquiry protocols. The Liverpool-Toulouse group proposed a protocol for general inquiry dialogues (e.g. [2, 9]), however this protocol can lead to unsuccessful dialogues in which no argument for the topic is found even when such an argument does exist in the union of the two agents’ beliefs. In [8], McBurney and Parsons present a specialised inquiry protocol for use in scientific domains, such as in assessments of carcinogenic risk of new chemicals, however this protocol is too complicated for general use, containing over thirty specialised moves. Neither of these groups have proposed a strategy for use with their inquiry protocol, i.e. their systems model inquiry dialogues but are not sufficient to generate them.

A key contribution of this work is that we not only provide a protocol for modelling inquiry dialogues but we also provide a specific strategy to be followed, making this system sufficient to also generate inquiry dialogues. Other works have also considered the automation of dialogues. For example, [12] gives an account of the different factors which must be considered when designing a dialogue strategy. The Liverpool-Toulouse group [9] explore the effect of different agent attitudes, which reduce the set of legal moves from which an agent must choose a move but do not select exactly one of the legal moves to make. Pasquier et al.’s cog-
nitive coherence theory [10] addresses the pragmatic issue of
dialogue generation, but it is not clear what behaviour
this would produce. Both [1] and [7] propose a formalism
for representing the private strategy of an agent to which
argumentation is then applied to determine the move to be
made at a point in a dialogue, however neither give a specific
strategy for inquiry dialogues.

The lack of specific dialogue strategies proposed in the
literature makes it hard to analyse the general behaviour of
dialogues. A specific strategy that has been proposed for
persuasion dialogues is in [3], but this is not formally stated
and no analysis of the dialogue behaviour is given. As far as
we are aware, ours is the only example of a system that incor-
porates a strategy capable of generating inquiry dialogues,
and this allows us to consider soundness and completeness
properties for our system.

2. MOTIVATION

Our work has been motivated by the medical domain. Ar-
gumentation allows us to deal with the incomplete, inconsis-
tent and uncertain knowledge that is characteristic of med-
cal knowledge. There are often many different healthcare
professionals involved in the care of a patient, each of whom
has a particular type of specialised knowledge and who must
cooperate in order to provide the best possible care for the
patient. For example, in hospitals in the UK it is usually the
case that there is a multi-disciplinary meeting each week to
discuss patients with breast cancer. There may be around
twenty different healthcare professionals at such a meeting
including, for example, nurses, radiographers, oncologists,
surgeons and even statisticians. The diagnosis and prognos-
sis of each of the current breast cancer patients are discussed
at such meetings and different treatment plans are consid-
ered. However, healthcare professionals do not always make
the correct decision given the available knowledge. Accord-
ing to various studies, a patient with breast cancer may be
up to seven times more likely to receive a cure at one of the
best specialist centres than at a general medical facility [5].

We wish to standardise the level of care provided to patients
by providing agent-based support.

Inquiry dialogues are a type of dialogue that would be of
particular use in the healthcare domain, where it is often the
case that people have distinct types of knowledge and so need to interact with others in order to have all the infor-
mation necessary to make a decision. Consider the situation
in which a consultant must diagnose a patient who has sus-
ppected cancer. The consultant will have lots of knowledge
about cancer and the different forms suggested by differ-
ent healthcare professionals. The diagnosis and prognosis
of each of the current breast cancer patients are discussed
at such meetings and different treatment plans are consid-
ered. However, healthcare professionals do not always make
the correct decision given the available knowledge. Accord-
ing to various studies, a patient with breast cancer may be
up to seven times more likely to receive a cure at one of the
best specialist centres than at a general medical facility [5].

We wish to standardise the level of care provided to patients
by providing agent-based support.

Inquiry dialogues are a type of dialogue that would be of
particular use in the healthcare domain, where it is often the
case that people have distinct types of knowledge and so need to interact with others in order to have all the infor-
mation necessary to make a decision. Consider the situation
in which a consultant must diagnose a patient who has sus-
ppected cancer. The consultant will have lots of knowledge
about cancer and the different forms suggested by differ-
ent healthcare professionals. The diagnosis and prognosis
of each of the current breast cancer patients are discussed
at such meetings and different treatment plans are consid-
ered. However, healthcare professionals do not always make
the correct decision given the available knowledge. Accord-
ing to various studies, a patient with breast cancer may be
up to seven times more likely to receive a cure at one of the
best specialist centres than at a general medical facility [5].

We wish to standardise the level of care provided to patients
by providing agent-based support.

Inquiry dialogues are a type of dialogue that would be of
particular use in the healthcare domain, where it is often the
case that people have distinct types of knowledge and so need to interact with others in order to have all the infor-
mation necessary to make a decision. Consider the situation
in which a consultant must diagnose a patient who has sus-
ppected cancer. The consultant will have lots of knowledge
about cancer and the different forms suggested by differ-
ent healthcare professionals. The diagnosis and prognosis
of each of the current breast cancer patients are discussed
at such meetings and different treatment plans are consid-
ered. However, healthcare professionals do not always make
the correct decision given the available knowledge. Accord-
ing to various studies, a patient with breast cancer may be
up to seven times more likely to receive a cure at one of the
best specialist centres than at a general medical facility [5].

We wish to standardise the level of care provided to patients
by providing agent-based support.

Inquiry dialogues are a type of dialogue that would be of
particular use in the healthcare domain, where it is often the
case that people have distinct types of knowledge and so need to interact with others in order to have all the infor-
mation necessary to make a decision. Consider the situation
in which a consultant must diagnose a patient who has sus-
ppected cancer. The consultant will have lots of knowledge
about cancer and the different forms suggested by differ-
ent healthcare professionals. The diagnosis and prognosis
of each of the current breast cancer patients are discussed
at such meetings and different treatment plans are consid-
ered. However, healthcare professionals do not always make
the correct decision given the available knowledge. Accord-
ing to various studies, a patient with breast cancer may be
up to seven times more likely to receive a cure at one of the
best specialist centres than at a general medical facility [5].

We wish to standardise the level of care provided to patients
by providing agent-based support.
We now define an argument as being a minimally consistent set from which the claim can be defeasibly derived.

**Definition 3.4.** An argument constructed from a set of, possibly inconsistent, beliefs \( \Psi (\Psi \subseteq \mathcal{B}) \) is a tuple \((\Phi, \phi)\) where \( \phi \) is a defeasible fact and \( \Phi \) is a set of beliefs such that:

1. \( \Phi \subseteq \Psi \),
2. \( \Phi \models \phi \),
3. \( \forall \phi, \phi' \text{ s.t. } \Phi \models \phi \text{ and } \Phi \models \phi' \), it is not the case that \( \phi \cup \phi' \vdash \bot \) (where \( \vdash \) represents classical implication),
4. there is no subset of \( \Phi \) that satisfies (1-3).

\( \phi \) is called the support of the argument and \( \phi \) is called the claim. The set of all arguments that can be constructed from a set of beliefs \( \Psi \) is denoted \( \mathcal{A}(\Psi) \).

In the following section we define our dialogue system for argument inquiry dialogues. This allows two agents to share beliefs in order to jointly construct arguments for a specific claim.

## 4. DIALOGUE SYSTEM

The communicative acts in a dialogue are called moves. We assume that there are always exactly two agents (participants) taking part in a dialogue, each with its own identifier taken from the set \( \mathcal{I} \). Each participant takes it in turn to make a move to the other participant. For a dialogue involving participants \( x_1, x_2 \in \mathcal{I} \), we also refer to participants using the meta-variables \( P \) and \( T \) such that if \( P = x_1 \) then \( T = x_2 \) and if \( P = x_2 \) then \( T = x_1 \).

A move in our system is of the form \((Agent, Act, Content)\). Agent is the identifier of the agent to which the move is addressed (the receiver of the move), Act is the type of move, and the Content gives the details of the move. The format for moves used in argument inquiry dialogues is shown in Table 1, and the set of all argument inquiry moves meeting the format defined in Table 1 is denoted \( \mathcal{M} \). Note that the system allows for other types of dialogues to be generated and these might require the addition of extra moves. Also, \( \text{Receiver}(\{Agent, Act, Content\}) = Agent \).

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 the other. Each move is indexed by the timepoint when the move was made. Exactly one move is made at each timepoint. The dialogue itself is indexed with two timepoints, indexing the first and last moves of the dialogue.

**Definition 4.1.** A dialogue, denoted \( D_r \), is a sequence of moves of the form \( \{m_1, \ldots, m_t\} \) involving two participants \( x_1, x_2 \) such that \( x_1, x_2 \in \mathcal{I} \), \( x_1 \neq x_2 \), \( r,t \in \mathbb{N} \) and the following conditions hold:

1. \( m_r \) is a move of the form \((P, open, \gamma)\)
2. \( \text{Receiver}(m_s) \in \{x_1, x_2\} \) \( (r \leq s \leq t) \)
3. \( \text{Receiver}(m_s) \neq \text{Receiver}(m_{s+1}) \) \( (r \leq s < t) \)

The topic of the dialogue \( D_r \) is returned by \( \text{Topic}(D_r) \) such that \( \text{Topic}(D_r) = \gamma \). The set of all dialogues is denoted \( \mathcal{D} \).

### Table 1: The format for moves used in argument inquiry dialogues, where \( \gamma \) is a belief, \( \langle \Phi, \phi \rangle \) is an argument and \( x \) is an agent \((x \in \mathcal{I})\).

<table>
<thead>
<tr>
<th>Move</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td>((x, open, \gamma))</td>
</tr>
<tr>
<td>assert</td>
<td>((x, assert, \langle \Phi, \phi \rangle))</td>
</tr>
<tr>
<td>close</td>
<td>((x, close, \gamma))</td>
</tr>
</tbody>
</table>

The first move of a dialogue \( D_r \) must always be an open move (condition 1 of the previous definition), every move of the dialogue must be made to a participant of the dialogue (condition 2), and the agents take it in turns to receive moves (condition 3).

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

**Definition 4.2.** Let \( D_r^1 \) and \( D_r^2 \) be two dialogues. \( D_r^1 \) is a sub-dialogue of \( D_r^2 \) if \( \text{ff} D_r^1 \) is a sub-sequence of \( D_r^2 \) \((r < r_1 \leq t_1 \leq T)\). \( D_r^1 \) is a top-level dialogue if \( r = 1 \). \( D_r^2 \) is a top-dialogue of \( D_r^1 \) if either the sequence \( D_r^1 \) is the same as the sequence \( D_r^2 \) or \( D_r^1 \) is a sub-dialogue of \( D_r^2 \). If \( D_r^1 \) is a sequence of \( n \) moves, \( D_r^2 \) extends \( D_r^1 \) if the first \( n \) moves of \( D_r^2 \) are the sequence \( D_r^1 \).

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

**Definition 4.3.** Let \( D_r^1 \) be a dialogue with participants \( x_1 \) and \( x_2 \) such that \( \text{Topic}(D_r^1) = \gamma \). We say that \( m_r \ (r < s \leq t) \) is a matched-close for \( D_r^1 \) if \( m_s = (P, close, \gamma) \) and \( m_t = (T, close, \gamma) \).

So a matched-close will terminate a dialogue \( D_r^1 \) but only if \( D_r^1 \) has not already terminated and any sub-dialogues that are embedded within \( D_r^1 \) have already terminated.

**Definition 4.4.** Let \( D_r^1 \) be a dialogue. \( D_r^1 \) terminates at \( t \) iff the following conditions hold:

1. \( m_r \) is a matched-close for \( D_r^1 \),
2. \( \neg \exists D_r^2 \text{ s.t. } D_r^2 \text{ terminates at } t_1 \) and \( D_r^1 \) extends \( D_r^2 \),
3. \( \forall D_r^1 \text{ if } D_r^1 \text{ is a sub-dialogue of } D_r^1 \), then \( \exists D_r^2 \text{ s.t. } D_r^2 \text{ terminates at } t_2 \) and either \( D_r^2 \text{ extends } D_r^1 \) or \( D_r^1 \text{ extends } D_r^2 \) and \( D_r^2 \) is a sub-dialogue of \( D_r^1 \).

As we are often dealing with multiple nested dialogues it is sometimes useful to refer to the current dialogue, which is the innermost dialogue that has not yet terminated.

**Definition 4.5.** Let \( D_r^1 \) be a dialogue. The current dialogue is returned by \( \text{Current}(D_r^1) \) such that \( \text{Current}(D_r^1) = D_r^1 \ (1 \leq r \leq r_1 \leq T) \) where the following conditions hold:

1. \( m_{s+1} = (x, open, \gamma) \) for some \( x \in \mathcal{I} \) and some \( \gamma \in \mathcal{B} \),
2. \( \forall D_r^2 \text{ if } D_r^2 \text{ is a sub-dialogue of } D_r^1 \), then \( \exists D_r^2 \text{ s.t. either } D_r^2 \text{ extends } D_r^1 \text{ or } D_r^1 \text{ extends } D_r^2 \text{ and } D_r^2 \text{ is a sub-dialogue of } D_r^1 \) and \( D_r^2 \) terminates at \( t_2 \),
3. \( \exists T_{d_{i_1}} s.t. T_{d_{i_1}} \) extends \( D_{t_{i_2}} \) and \( D_{t_{i_2}} \) terminates at \( t_3 \).

The topic of the current dialogue is returned by the function \( cTopic(D_t) \) such that \( cTopic(D_t) = \text{Topic}(Current(D_t)) \).

A schematic example of nested argument inquiry dialogues is now given. The top-level dialogue is \( D_t^1 \), such that:

\[
D_t^1 = [m_1, \ldots, m_t, m_{t-1}, m_t, \ldots, m_k]
\]

where

\[
m_1 = (P_k, \text{open}, \gamma_k) \quad m_t = (P_t, \text{open}, \gamma_t)
\]

\[
m_{t-1} = (P_{t-1}, \text{close}, \gamma_{t-1})
\]

\[
m_k = (P_k, \text{close}, \gamma_k)
\]

\[
m_t = (P_t, \text{close}, \gamma_t)
\]

\( D_t^k \) has not yet terminated and is the current dialogue. There are two sub-dialogues of \( D_t^k \), the first is \( D_t^k \) and this terminates at \( t \).

\[
D_t^k = [m_i, \ldots, m_j, m_{j-1}, m_j, \ldots, m_{t-1}, m_t]
\]

The second is \( D_t^k = [m_j, \ldots, m_{j-1}, m_k] \), which terminates at \( k \).

We adopt the standard approach of associating a commitment store with each agent participating in a dialogue. A commitment store is a set of beliefs that the agent has asserted so far in the course of the dialogue. As a commitment store consists of things that the agent has already publicly declared, its contents are visible to the other agent participating in the dialogue. For this reason, when constructing an argument, an agent may make use of not only its own beliefs but also those from the other agent’s commitment store.

Definition 4.6. A commitment store associated with an agent \( x \) at a timepoint \( t \), denoted \( CS^x_t \), where \( x \in I \) and \( t \in N \), is a set of beliefs (i.e. \( CS^x_t \subseteq B \)).

An agent’s commitment store grows monotonically over time. If an agent makes a move asserting an argument, every element of the support is added to the agent’s commitment store. This is the only time the commitment store is updated.

Definition 4.7. Commitment store update. Let the current dialogue be \( D_t^k \) with participants \( x_1 \) and \( x_2 \):

\[
CS^x_t = \begin{cases}
\emptyset & \text{if } t = 0, \\
CS^{x_1}_{t-1} \cup \Phi & \text{if } m_t = (T, \text{assert}, \langle \Phi, \phi \rangle), \\
CS^{x_1}_{t-1} & \text{otherwise}
\end{cases}
\]

The goal of an argument inquiry dialogue is for a pair of agents to jointly construct an argument for a particular claim, \( \phi \). So the first move of a top-level argument inquiry dialogue is an open move with the defeasible fact \( \phi \) as its content. Now, if one agent knows of a defeasible rule \( \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi \), then it can make a move to open a nested argument inquiry dialogue with \( \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi \) as its topic.

If the two participants could jointly provide an argument for each literal \( \alpha_i \) (\( 1 \leq i \leq n \)) in the antecedent of the topic, then it would be possible to construct an argument for \( \phi \). We keep track of this set of literals in a question store: so when an argument inquiry dialogue with topic \( \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi \) is opened, a question store associated with that dialogue is created whose contents are \( \{\alpha_1, \ldots, \alpha_n\} \). Throughout the dialogue the participating agents will both try to provide arguments for the literals in the question store. This may lead them to open further nested argument inquiry dialogues that have as a topic a rule whose consequent is a literal in the current question store.

Definition 4.8. Let \( \gamma \) be a defeasible fact. Let \( \delta \) be a defeasible rule of the form \( \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta \). For a dialogue \( D_t^k \) with participants \( x_1 \) and \( x_2 \), a question store, denoted \( QS_{x_i} \), is a finite set of literals such that:

\[
QS_{x_i} = \begin{cases}
\{ \gamma \} & \text{if } m_t = (P, \text{open}, \gamma), \\
\{ \alpha_1, \ldots, \alpha_n \} & \text{if } m_t = (P, \text{open}, \delta), \\
\emptyset & \text{otherwise}
\end{cases}
\]

The question store of the current dialogue is returned by \( cQS \) such that \( cQS(D_t^k) = QS_{x_i} \) if \( \text{Current}(D_t^k) = D_t^k \).

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 argument inquiry 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 4.9. The argument inquiry protocol is a function \( \Pi : D \times I \rightarrow \varphi(M) \). If \( D_t^k \) is a well-formed, top-level dialogue with participants \( x_1 \) and \( x_2 \) such that Receiver(\( m_t \)) = \( P \), \( 1 \leq t \) and \( cTopic(D_t^k) = \gamma \), then \( \Pi(D_t^k, P) \) is:

\[
\Pi_{assert}(D_t^k, P) \cup \Pi_{open}(D_t^k, P) \cup \{ (P, \text{close}, \gamma) \}
\]

\[
\Pi_{assert}(D_t^k, P) = \{ (T, \text{assert}, \langle \Phi, \phi \rangle) \}
\]

\[
\Pi_{open}(D_t^k, P) = \{ (T, \text{open}, \beta_1 \wedge \ldots \wedge \beta_n \rightarrow \alpha) \}
\]

\[
\Pi_{assert}(D_t^k, P) \cup \Pi_{open}(D_t^k, P) \cup \{ (P, \text{close}, \gamma) \}
\]

where

\[
\Pi_{assert}(D_t^k, P) = \{ (T, \text{assert}, \langle \Phi, \phi \rangle) \}
\]

\[
\Pi_{open}(D_t^k, P) = \{ (T, \text{open}, \beta_1 \wedge \ldots \wedge \beta_n \rightarrow \alpha) \}
\]

Note that it is straightforward to check conformance with the protocol as the protocol only refers to public elements of the dialogue.

We will shortly give a specific strategy function that allows an agent to select exactly one legal move to make at each timepoint in an argument inquiry dialogue. A strategy is personal to an agent and the move that it returns depends on the agent’s private beliefs. The argument inquiry strategy states that if there are any legal moves that assert an argument that can be constructed by the agent then a single one of these moves is selected (according to a selection function that we define shortly denoted \( \text{Pick}_k \)), else if there are any legal open moves with a defeasible rule as their content that is in the agent’s beliefs then a single one of these moves is selected (according to a selection function that we define as \( \text{Pick}_k \)). If there are no such moves then a close move is made.
\[ \Omega(D_i^j, P) = \begin{cases} \text{Pick}_k(\text{Assert}(D_i^j, P)) & \text{iff } \text{Asserts}(D_i^j, P) \neq \emptyset \\ \text{Pick}_k(\text{Open}(D_i^j, P)) & \text{iff } \text{Asserts}(D_i^j, P) = \emptyset \text{ and } \text{Opens}(D_i^j, P) \neq \emptyset \\ \{P, \text{close}, c\text{Topic}(D_i^j)\} & \text{iff } \text{Asserts}(D_i^j, P) = \emptyset \text{ and } \text{Opens}(D_i^j, P) = \emptyset \end{cases} \]

where \( \text{Asserts}(D_i^j, P) = \{ (P, \text{assert}, (\Phi, \phi)) \in \Pi_{\text{assert}}(D_i^j, P) \mid (\Phi, \phi) \in \mathcal{A}(\Sigma^P \cup CS^P) \} \)

\( \text{Opens}(D_i^j, P) = \{ (P, \text{open}, \emptyset) \in \Pi_{\text{open}}(D_i^j, P) \mid \emptyset \in \Sigma^P \} \)

![Figure 1: The argument strategy function.](image)

In order to select a single one of the legal assert or open moves, we assign a unique number to the move content and carry out a comparison of these numbers. Let us assume that \( B \) is composed of a finite number \( Z \) of atoms. Let us also assume that there is a registration function \( \mu \) over these atoms. For a rule \( L_1 \land \ldots \land L_n \rightarrow L_{n+1} \), \( \mu(L_1 \land \ldots \land L_n \rightarrow L_{n+1}) \) is an \( n+1 \) digit number of the form \( \mu(L_1) \cdot \mu(L_n) \cdot \mu(L_{n+1}) \). This gives us a unique base \( Z \) number for each formula in \( B \) and allows us to choose a unique open move.

**Definition 4.10.** Consider the set of open moves \( \Psi = \{(P, \text{open}, \phi_1), \ldots, (P, \text{open}, \phi_k)\} \). The function \( \text{Pick}_k \) returns the chosen open move. \( \text{Pick}_k(\Psi) = (P, \text{open}, \phi_i) \) (1 \( \leq i \leq k \)) such that for all \( j \) (1 \( \leq j \leq k \)) if \( i \neq j \), then \( \mu(\phi_i) < \mu(\phi_j) \).

We can similarly assign a number to each argument in \( \mathcal{A}(B) \) using a registration function \( \lambda \) together with \( \mu \). For an argument \( \langle \phi_1, \ldots, \phi_n, n+1 \rangle \),

\[ \lambda(\langle \phi_1, \ldots, \phi_n, n+1 \rangle) = \langle d_1, \ldots, d_n, d_{n+1} \rangle \]

where \( d_1 < \ldots < d_n < d_{n+1} \) and \( \langle d_1, \ldots, d_n, d_{n+1} \rangle \) is a permutation of \( \mu(\phi_1), \ldots, \mu(\phi_n), \mu(\phi_{n+1}) \) (where \( \mu \) is the registration function for \( B \)). The function \( \lambda \) returns a unique tuple of base \( Z \) numbers for each argument. We use a standard lexicographical comparison, denoted \( \preceq \), of these tuples of numbers to choose a unique assert move.

**Definition 4.11.** Consider the set of assert moves \( \Psi = \{(P, \text{assert}, \langle\Phi, \phi_1\rangle), \ldots, (P, \text{assert}, \langle\Phi_0, \phi_n\rangle)\} \). The function \( \text{Pick}_k \) returns the chosen assert move. \( \text{Pick}_k(\Psi) = (P, \text{assert}, \langle\Phi_i, \phi_i\rangle) \) (1 \( \leq i \leq k \)) such that for all \( j \) (1 \( \leq j \leq k \)) if \( i \neq j \), then \( \lambda(\langle\Phi_i, \phi_i\rangle) \preceq \lambda(\langle\Phi_j, \phi_j\rangle) \).

We now use these functions to define the argument inquiry strategy. 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 exactly one of the legal moves.

**Definition 4.12.** The argument inquiry strategy is a function \( \Omega : D \times I \rightarrow M \) as given in Figure 1.

Note that a top-level argument inquiry dialogue will always have a defeasible fact as its topic, but each of its sub-dialogues will always have a defeasible rule as its topic. The argument inquiry strategy constrains an agent to only opening nested sub-dialogues that have as their topic a defeasible rule that is present in the agent’s belief base. This prevents an agent from opening a nested sub-dialogue unless it at least knows of a rule that might help construct a desired argument.

We are now able to define a well-formed argument inquiry dialogue. This is a dialogue that does not continue after it has terminated and that is generated by the argument inquiry strategy.

**Definition 4.13.** Let \( D_i^j \) be a dialogue with participants \( x_1 \) and \( x_2 \). \( D_i^j \) is a well-formed argument inquiry dialogue if the following conditions hold:

1. \( \neg \exists t' \ (r < t' < t) \text{ s.t. } D_i^j \text{ extends } D_i^{t'} \text{ and } D_i^{t'} \text{ terminates at } t' \).
2. \( \forall s \ (r < s < t) \text{ s.t. } D_i^j \text{ extends } D_i^s \text{, if } D_i^s \text{ is a top-dialogue of } D_i^j \text{ and } D_i^s \text{ extends } D_i^r \text{ and } D_i^r \text{ extends } D_i^s \text{ then } \text{Receiver}(m_i) = P \) (where \( P \in \{x_1, x_2\} \)),

An example of a well-formed argument inquiry dialogue with participants \( x_1 \) and \( x_2 \) is shown in Table 2. The first column of the table gives the value of \( t \), the second column gives the commitment store of agent \( x_1 \), the third column gives the move \( m_i \), the fourth column gives the commitment store of the agent \( x_2 \) and the fifth column gives the contents of any question stores that are not equal to the empty set. The agents’ belief bases are shown at the top of the table. We have highlighted nested sub-dialogues by drawing a box around them. Note that we assume some higher-level planning component that guides the agent when deciding whether to enter into a dialogue, who this dialogue should be with and on what topic, i.e. that makes the decision to make the move \( m_1 \).

In our simple example, it seems that it would be more straightforward to pool both agents beliefs and apply a reasoning procedure to this set of beliefs. However, given a real-world scenario this would not necessarily be the case. When dealing with the medical domain we have to consider privacy issues that would restrict agents from simply pooling all beliefs. It is also sometimes the case that agents have vast belief bases and the communication cost involved in sharing all beliefs would be prohibitive.

**5. SOUNDNESS AND COMPLETENESS**

We believe that it is important to consider soundness and completeness properties if we are to understand the behaviour of our dialogues. This is particularly the case when
The goal of the dialogue, which is successfully achieved, is to find an argument for the set of all arguments that can be constructed from the union of the commitment stores and whose claims are in the question store. We define the outcome of an argument inquiry dialogue as the set of all arguments that can be constructed from the union of the commitment stores and whose claims are in the question store.

**Definition 5.1.** The argument inquiry outcome of a dialogue is a function \( \text{Outcome} : T \rightarrow \phi(A(B)) \). If \( D \) is a well-formed argument inquiry dialogue with participants \( x_1 \) and \( x_2 \), then \( \text{Outcome}(D) = \{ (\Phi, \phi) \in A(CS_{x_1} \cup CS_{x_2}) \mid \phi \in QS \} \).

The goal of an argument inquiry dialogue is for the two agents to share appropriate parts of their knowledge in order to try to construct an argument for a specific claim or claims. The benchmark that we compare the outcome of the dialogue with is the set of arguments that can be constructed from the union of the two agents’ beliefs. So this benchmark is, in a sense, the ‘ideal situation’ where there are clearly no constraints on the sharing of beliefs.

As far as we are aware, the only other similar work that considers soundness and completeness properties is [13]. In [13] Sadri et al. define different agent programs for negotiation. If such an agent program is both exhaustive and deterministic then exactly one move is suggested by the program at a timepoint, making such a program generative and allowing consideration of soundness and completeness properties. Since other inquiry dialogue systems do not provide a specific strategy, they miss the chance to better understand the dialogue behaviour by considering such properties.

We say that an argument inquiry dialogue is sound if and only if, when the outcome of the dialogue includes an argument, then that same argument can be constructed from the union of the two participating agents’ beliefs.

**Definition 5.2.** Let \( D \) be a well-formed argument inquiry dialogue with participants \( x_1 \) and \( x_2 \). We say that \( D \) is sound if, if \( (\Phi, \phi) \in \text{Outcome}(D) \), then \( (\Phi, \phi) \in A(S_1 \cup S_2) \).

In order to show that all argument inquiry dialogues are sound we need to introduce a few lemmas. Please note that we have omitted the more obvious proofs from this presentation but for a thorough discussion of all results please refer to the first author’s thesis [4].

The first states that if an agent asserts an argument, then it must be able to construct the argument from its beliefs and the other agent’s commitment store. This is clear from the definition of the argument inquiry strategy.

**Lemma 5.1.** Let \( D \) be a well-formed argument inquiry dialogue with participants \( x_1 \) and \( x_2 \). If \( D \) is a top-dialogue of \( D \), then \( (P, \text{assert}, (\Phi, \phi)) \) is a top-dialogue of \( A(S_1 \cup S_2) \).

From Lemma 5.1 and the fact that the commitment stores are only updated when an assert move is made, we get the following.

<table>
<thead>
<tr>
<th>( \Sigma^x = {d, b \rightarrow c} )</th>
<th>( m_1 )</th>
<th>( \Sigma^x = {e, d \land e \rightarrow b, a \rightarrow b} )</th>
<th>( Q )</th>
<th>( Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
<td>( CS_{x_1} )</td>
<td>( m_1 )</td>
<td>( CS_{x_2} )</td>
<td>( QS )</td>
</tr>
<tr>
<td>1</td>
<td>( {x_1, \text{open}, c} )</td>
<td>( QS_1 = {c} )</td>
<td>( QS_2 = {b} )</td>
<td>( QS_3 = {a} )</td>
</tr>
<tr>
<td>2</td>
<td>( {x_2, \text{open}, b \rightarrow c} )</td>
<td>( QS_2 = {b} )</td>
<td>( QS_3 = {a} )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>3</td>
<td>( {x_1, \text{open}, a \rightarrow b} )</td>
<td>( QS_3 = {a} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>4</td>
<td>( {x_2, \text{close}, a \rightarrow b} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>5</td>
<td>( {x_1, \text{close}, a \rightarrow b} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>6</td>
<td>( {x_2, \text{close}, b \rightarrow c} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>7</td>
<td>( {x_1, \text{open}, d \land e \rightarrow b} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>8</td>
<td>( {x_2, \text{assert}, {d}, {d}} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>9</td>
<td>( {x_1, \text{assert}, {e}, {e}} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>10</td>
<td>( {x_2, \text{close}, d \land e \rightarrow b} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>11</td>
<td>( {x_1, \text{close}, d \land e \rightarrow b} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>12</td>
<td>( {x_2, \text{close}, b \rightarrow c} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>13</td>
<td>( {x_1, \text{assert}, {d, e, d \land e \rightarrow b}, {b}} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>14</td>
<td>( {x_2, \text{close}, b \rightarrow c} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>15</td>
<td>( {x_1, \text{close}, b \rightarrow c} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>16</td>
<td>( {x_2, \text{assert}, {d, e, d \land e \rightarrow b, b \rightarrow c}, {c}} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>17</td>
<td>( {x_1, \text{close}, c} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
<tr>
<td>18</td>
<td>( {x_2, \text{close}, c} )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
<td>( QS_4 )</td>
</tr>
</tbody>
</table>
Lemma that a commitment store is always a subset of the union of the two agents’ beliefs.

**Lemma 5.2.** If $D_i^t$ is a well-formed argument inquiry dialogue with participants $x_1$ and $x_2$, then $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma_1^{t+1} \cup \Sigma_2^{t+2}$.

The next lemma states that if we have a set $\Phi$ that is a subset of a set of beliefs $\Psi$, then the set of arguments that can be constructed from $\Phi$ is a subset of the set of arguments that can be constructed from $\Psi$.

**Lemma 5.3.** Let $\Phi \subseteq B$ and $\Psi \subseteq B$ be two sets. If $\Phi \subseteq \Psi$, then $A(\Phi) \subseteq A(\Psi)$.

We now show that argument inquiry dialogues are sound.

**Theorem 5.1.** If $D_i^t$ is a well-formed argument inquiry dialogue with participants $x_1$ and $x_2$, then $D_i^t$ is sound.

**Proof:** Assume $\langle \Phi, \phi \rangle \in \text{Outcome}(D_i^t)$. From Def. 5.1, $\langle \Phi, \phi \rangle \in A(CS_{x_1}^t \cup CS_{x_2}^t)$. From Lem. 5.2 $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma_1^{t+1} \cup \Sigma_2^{t+2}$. Hence, from Lem. 5.3, $\Phi \in A(\Sigma_1^{t+1} \cup \Sigma_2^{t+2})$. Hence, $D_i^t$ is sound. \( \square \)

Similarly, an argument inquiry dialogue is complete if and only if, if the dialogue terminates at $t$ and it is possible to construct an argument for a literal in the question store from the union of the two participating agents’ beliefs, then that argument will be in the outcome of the dialogue at $t$.

**Definition 5.3.** Let $D_i^t$ be a well-formed argument inquiry dialogue with participants $x_1$ and $x_2$. We say that $D_i^t$ is complete if, if $D_i^t$ terminates at $t$, and there exists $\phi$ such that $\langle \Phi, \phi \rangle \in A(\Sigma_1^{t+1} \cup \Sigma_2^{t+2})$, then $\langle \Phi, \phi \rangle \in \text{Outcome}(D_i^t)$.

In order to show that all argument inquiry dialogues are complete we need to give some further lemmas. The first states that if an argument inquiry dialogue terminates at $t$, then the set of legal moves from which an agent must choose the move $m_i$ does not include any open or assert moves. This is clear from the definition of the argument inquiry strategy.

**Lemma 5.4.** If $D_i^t$ is a well-formed argument inquiry dialogue that terminates at $t$ with participants $x_1$ and $x_2$ such that $\text{Receiver}(m_{i-1}) = P$ and $D_i^t$ extends $D_{i-1}^t$ and $D_{i-1}^t$ is a top-dialogue of $D_{i-1}^t$, then $\text{Asserts}(D_{i-1}^t, P) = \emptyset$ and $\text{Opens}(D_{i-1}^t, P) = \emptyset$.

From Lemmas 5.4 and 5.5 and the definitions of argument inquiry strategy and the argument inquiry protocol we get the following lemma that if two agents are in an argument inquiry dialogue that terminates at $t$ and there exists some $r$ ($1 \leq r < t$) such that $\phi \in QS_r$ and there is an argument for $\phi$ of the form $\{\phi\}$, then $\phi$ can be constructed from the union of the two agents’ beliefs, then $\phi$ will be in the union of the commitment stores at timepoint $t$.

**Lemma 5.5.** For all $r$ ($1 \leq r < t$), if $D_i^t$ is a well-formed argument inquiry dialogue that terminates at $t$ with participants $x_1$ and $x_2$ such that $\phi \in QS_r$ and there is an argument $\{\phi\} \in A(\Sigma_1^{t+1} \cup \Sigma_2^{t+2})$, then $\phi \in CS_{x_1}^t \cup CS_{x_2}^t$.

From Lemmas 5.4 and 5.5 and the definitions of argument inquiry strategy and the argument inquiry protocol we also get the following lemma that if there is a defeasible rule whose consequent is present in the question store, then there will be a timepoint at which a question store will be created that contains all the literals of the antecedent of the defeasible rule.

**Lemma 5.6.** For all $r$ ($1 \leq r < t$), if $D_i^t$ is a well-formed argument inquiry dialogue that terminates at $t$ with participants $x_1$ and $x_2$ such that $\phi \in QS_r$ and there exists a defeasible rule $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi \in \Sigma_1^{t+1} \cup \Sigma_2^{t+2}$, then there exists $t_1$ ($1 < t_1 < t$) such that $QS_{t_1} = \{\alpha_1, \ldots, \alpha_n\}$ and $D_i^t$ extends $D_i^{t_1}$.

We now show that argument inquiry dialogues are complete.

**Theorem 5.2.** If $D_i^t$ is a well-formed argument inquiry dialogue with participants $x_1$ and $x_2$, then $D_i^t$ is complete.

**Proof:** Assume $D_i^t$ does not terminate at $t$ then $D_i^t$ is complete. So assume $D_i^t$ terminates at $t$, $\phi \in QS_r$, and $\langle \Phi, \phi \rangle \in A(\Sigma_1^{t+1} \cup \Sigma_2^{t+2})$. By Def. 3.4, $\Phi \subseteq \Sigma_1^{t+1} \cup \Sigma_2^{t+2}$. There are two cases. (Case 1) $\Phi = \{\phi\}$. Hence from Lem. 5.5, $\phi \in CS_{x_1}^t \cup CS_{x_2}^t$. From Def. 5.1, $\langle \Phi, \phi \rangle \in \text{Outcome}(D_i^t)$. (Case 2) $\exists \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi \in \Phi$. By Def. 3.4, $\exists \alpha \in \Phi$ such that $\langle \Phi, \alpha \rangle \in A(\Sigma_1^{t+1} \cup \Sigma_2^{t+2})$. From Lem. 5.6, $\exists \alpha \subseteq \{\alpha_1, \ldots, \alpha_n\}$. Each $\Phi$ is either an example of case 1 or case 2, so, by recursion, $\exists \alpha_2, t_2 (r < r_2 < t_2 \leq t)$, such that $\langle \Phi, \alpha_2 \rangle \in \text{Outcome}(D_{i_2}^t)$. Hence, from Def. 3.4, $\langle \Phi, \alpha_2 \rangle \in \text{Outcome}(D_i^t)$. \( \square \)

The previous result is particularly interesting if we know that an argument inquiry dialogue terminates. Fortunately, we can show that all argument inquiry dialogues terminate (as agents’ belief bases are finite, hence there are only a finite number of assert and open moves that can be generated and agents cannot repeat these moves).

**Theorem 5.3.** For any well-formed argument inquiry dialogue $D_i^t$, there exists a $t_1$ ($r < t_1 \leq t$) such that $D_i^t$ terminates at $t_1$ and $D_i^{t_1}$ extends $D_i^r$.

From Theorems 5.2 and 5.3, we get the desired result that if an argument can be constructed from the union of the two participating agents’ beliefs whose claim is a literal from the current question store, then there will come a timepoint at which that argument is in the outcome of the dialogue.

**Theorem 5.4.** Let $D_i^t$ be a well-formed argument inquiry dialogue with participants $x_1$ and $x_2$. If $\phi \in QS_r$ and there exists $\Phi$ such that $\langle \Phi, \phi \rangle \in A(\Sigma_1^{t+1} \cup \Sigma_2^{t+2})$, then there exists $t_1$ ($1 < t_1 < t$) such that $D_i^t$ extends $D_i^{t_1}$ and $\langle \Phi, \phi \rangle \in \text{Outcome}(D_i^{t_1})$.

To summarise our results, we have shown that all argument inquiry dialogues terminate and that when an argument inquiry dialogue does terminate the set of arguments which makes up its outcome will be exactly the same as the set of all arguments that can be constructed from the union of the participating agents’ beliefs and that have as their claim a literal that is in the question store.

**6. FUTURE WORK**

In future work we would like to look at relaxing some of the assumptions we have made. As this is the first piece of
work examining soundness and completeness properties of inquiry dialogues, it is only a stepping stone on the way to being able to provide agents in the medical domain with the capability to fully support interactions such as the multi-disciplinary meeting discussed in Section 2. We would like to allow more than two agents to take part in an argument inquiry dialogue. There are often more than two individuals involved in making a decision in the medical domain and we would like our system to be able to deal with this.

We currently assume that an agent’s belief base does not change during a dialogue, and would like to consider the implications of dropping this assumption. It is likely that an agent may be carrying out several tasks at once and may even be involved in several different dialogues at once, and as a result it may be regularly updating its beliefs. This raises several questions. For example, if an agent’s belief base kept growing during a dialogue, would it be possible to generate infinite dialogues? And what should an agent do if it has cause to remove a belief from its belief base that it asserted earlier in the dialogue?

We would also like to further explore the benchmark which we compare our dialogue outcomes to. We currently compare the outcome that two agents participating in a dialogue arrive at with that which would be arrived at by a single agent that had as its beliefs the union of the participating agents’ beliefs. This seems like an ideal situation as there are no constraints on the sharing of beliefs. However, it is only ideal if we accept that the agents each have the same level of expertise regarding the beliefs. Consider the situation in which a medical student is discussing a diagnosis with a consultant. In this situation, the ideal benchmark might be the outcome that the consultant would reach without taking into account any of the student’s beliefs. Or there may be a situation in which we wish our argument inquiry dialogues not to produce every argument for a certain claim but only those which are considered to be the ‘best’ in some sense.

We have not presented our warrant inquiry dialogue here but we have found that argument inquiry dialogues are frequently embedded within warrant inquiry dialogues. We believe that, as one of the more simple dialogue types, it is particularly useful to embed argument inquiry dialogues within other dialogues, particularly when the agents are co-operating to some degree. We would like to further investigate the utility of argument inquiry dialogues when embedded in dialogues of different types.

7. CONCLUSIONS

We have presented a dialogue system and given details of a specific protocol and strategy for generating inquiry dialogues between two agents. This system is intended for use in a cooperative domain where we wish the results of a dialogue to be predetermined, such as the medical domain. Other groups have presented protocols capable of modelling inquiry dialogues (e.g. [9, 8]), however none have provided the means to select exactly one legal move at each timepoint. We have addressed this problem by providing a strategy function that selects exactly one move from the set of legal moves returned by the protocol. We have proposed a benchmark against which to compare the outcome of our dialogues, being a single agent reasoning with the union of the participating agents’ beliefs, and have shown that dialogues generated by our system are always sound and complete in relation to this benchmark. No other group has considered such properties of inquiry dialogues. We have only given details relating to argument inquiry dialogues but our system is capable of dealing with dialogues of different types (e.g. warrant inquiry dialogues) with the definition of alternative moves, protocols, strategies and outcome functions.

8. REFERENCES