Bilateral Agent Negotiation With Information-Seeking

Adil Hussain and Francesca Toni

Department of Computing, Imperial College London,
180 Queen's Gate, London SW7 2AZ, UK
{ah02, ft}@doc.ic.ac.uk

Abstract. We present a generative framework for bilateral agent negotiation that allows for information-seeking between agents in a resource re-allocation setting. In this framework each agent begins with beliefs as to which resources it has and desires for resources that it would like to have. We define the rules of the dialogues specifying the permissible messages, turn-taking and the order in which messages can be sent. The participants of the dialogues (i.e. the agents) adopt internal policies that enable them to conform to the dialogue rules whilst pursuing their individual desires.

1 Introduction

We present a framework for bilateral negotiation between agents in a resource re-allocation setting that allows for the sharing of information. In our framework agents begin with beliefs and desires as to which resources they have and would like to have respectively, and the resource re-allocation problem is that of distributing the resources in such a way that the agents fulfil their desires (i.e. get the resources they need).

We adopt a distributed approach to the resource re-allocation problem, such that the resources are exchanged by the agents themselves rather than relying on a central entity. The resource exchanges are made as a result of communication between the agents, in the form of dialogues, as defined in [6]. Two types of dialogues are modelled in this paper: information-seeking and negotiation. An information-seeking dialogue is used by an agent to add to its knowledge-base, for example, to determine which agents currently hold the resources that it desires or to learn of the desires of other agents. This is useful in the larger context of negotiation wherein agents exchange resources to achieve their individual desires.

We define the rules for the information-seeking and negotiation dialogues by specifying the permissible messages, turn-taking and the order in which messages can be sent. As well as defining the rules of the dialogues, which dictate what the possible legal messages are at different points in the dialogue, we also provide a means for specifying agent strategies (policies) that are able to select which one of the possible messages to send. The policies, internal to the agents, are modelled by means of preferences and action rules that specify conditions under which an action (not always the sending of a message) can be performed and the consequences of performing that action.

As in [1] and [4], our agent model is generative and makes use of commitments[1], which are taken on by an agent as a result of sending messages. However, our focus is on information-seeking and negotiation with the aim of exchanging resources, whereas the focus in [1] and [4] is on sharing knowledge in order to contruct arguments for specific propositional claims. Further, although our dialogues are bilateral, we do not limit our agent systems to two agents.

Our approach to modelling the dialogues builds upon the work of [5]. As in [5], we present a formal, logic-based approach to one-to-one agent negotiation for resource re-allocation, in the context of goal achievement in systems of agents with limited resource availability. However, as well as allowing information-seeking dialogues to accompany the negotiation, we allow for more complicated policies based on the knowledge of other agents' beliefs and desires. Further, the dialogue constraints based on which the dialogues of [5] are modelled are extended to include consequences, as in [4], which specify changes that the agent must make to its internal state as a result of performing an action.

As discussed in [3, 6], dialogues are most interesting in combination, for example, when dialogues run in parallel, in sequence or embedded within one another. The framework presented in this paper follows on from this discussion and allows for generative agent policies to be defined such that two types of dialogue, information-seeking and negotiation, can be combined to represent interactions more complex than would be possible with each dialogue alone.

The paper is structured as follows: Section 2 introduces the internal components of our agents. Section 3 defines the rules for the information-seeking and negotiation dialogues, as considered in this paper. Section 4 presents an example policy that demonstrates how agents may participate in the dialogues, i.e., how and when the agents send and respond to messages. Section 5 presents an example agent system based on the agent policy of the previous section and demonstrates the sequences of dialogues that ensue. Section 6 discusses properties of the agent policy and the resulting dialogues. Lastly, Section 7 summarises and sets up a research agenda highlighting possible directions for future work.

2 Agent Components

The agents in our system are made up of eight components: beliefs, desires, actions, messages, commitments, action rules, the evaluation mechanism and preferences, all explained below. There is no intention component as found in the BDI approach that guides what the agent does since this is covered implicitly by the other components, as explained later.

The literals that occur in this paper are to be read as follows: A literal starting with lower-case is grounded, a literal starting with upper-case is a variable and an underscore symbol (_) represents an anonymous variable for which the value does not matter. The symbol $\neg$ represents negation.

Beliefs. An agent can have a number of resources in its possession at any one time, and is initiated with beliefs as to which resources it initially has. Such

[1] Commitments are slightly different in [4], where they are referred to as shared beliefs.

beliefs, denoted B, are represented using a single predicate "has", as follows: $B(x){:}has(x, r)$. This is read as: "Agent x believes that agent x has resource r".[2] It is assumed that if an agent has such a belief then it does indeed have that resource. The resources that the agent has, and hence its belief-base, can change as a result of a negotiation dialogue. Beliefs about other agents' resources of the form $B(x){:}has(y, r)$ are ignored in this paper, such knowledge is captured instead by means of commitments, as explained later in this section.

Desires. An agent is initialised with desires as to which resources it would like to have. These desires, denoted D, are represented using a single predicate "has", as follows: $D(x){:}has(x, r)$. This is read as: "Agent x has a desire for agent x to have resource r".[3] Desires of the form $D(x){:}has(y, r)$ (i.e. "Agent x has a desire for agent y to have resource r") are ignored in this paper and left for future work.

Actions. An agent performs actions, according to the state of its internal components, with the overall aim of fulfilling its desires (i.e. obtaining the desired resources). The actions that we consider in this paper are of two types: external communicative actions (i.e. speech acts), which we term *send*, and internal actions, which we term *process*. Both types of actions have consequences that, as a result of performing, alter the internal state of the agent, as will be explained when Action Rules are discussed later in this section. The difference between the two types of action is that external actions are those which result in a speech act (i.e. a message) being sent to another agent; this may be to initiate a dialogue or to respond to a previously received message. Actions are represented as follows: $Type[Message]$, where $Type$ is either *send* or *process*, and $Message$ is the message to be sent or processed respectively. The form that $Message$ takes is explained below.

Messages. Dialogues are based on a sequence of messages between the participating agents, in alternation. Each agent has associated with it a message store that stores the messages received from other agents. A message, denoted M, is kept in the message store until such time that it can be processed and removed. A queue structure is assumed in the message store such that it is possible to distinguish messages by the order in which they are received. This is useful in determining which message to respond to first, where the agent policy allows a choice and time of receipt matters.

Messages are of the form: $\mu(X, Y, \overline{Z})$, where μ is the utterance, X is the agent sending the message, Y is the intended recipient of the message and $\overline{Z}$ is the content of the message, which may consist of more than one term. A concrete instance of a message is $offer(x, y, banana, apple)$, where $offer$ is the utterance, x is the agent sending the message, y is the intended recipient of the message and $(banana, apple)$ is the content of the message, the meaning of which will be explained when the dialogues are discussed in Section 3.

[2] Note that we do not assume a modal approach; a belief is treated as a fact at the level of representation and beliefs abouts beliefs are not permitted.

[3] As with beliefs, desires are treated as facts at the level of representation.

Commitments. Every agent has associated with it a commitment store that records some of the dialogue utterances made by it to other agents and by other agents to it. Thus each agent's commitment store is partitioned into parts: one part for each agent with which commitments are shared. The commitments, denoted C, are of three types: commitment to belief, commitment to desire and commitment to dialogue, all explained below. Note that the commitment store is internal to the agent and not visible to any other agent.

Commitments may be retracted. This prevents an agent from having inconsistent commitments and is useful, for example, when an agent's internal state changes such that it is inconsistent with a commitment it has to another agent. Retraction of a commitment is done by removing the commitment from the commitment store and sending a message of retraction to the agents to whom the agent has the commitment. The message of retraction takes the following form: $retract(X, Y, Commitment)$, where *retract* is the utterance, X is the sender of the message (and owner of the commitment), Y is the recipient of the message (and the agent to whom the commitment is held), and *Commitment* is the commitment being retracted, the form of which differs depending on the commitment, as explained below. In sending and receiving the message of retraction, the sender and receiver have an obligation to remove the commitment from their commitment stores.

Commitment to Belief. As part of a dialogue sequence, if an agent (say x) notifies another agent (say y) of one of its beliefs (say $B(x){:}has(x,r)$), then, depending on its internal policy, x may store this as a commitment to the recipient agent (y), as follows: $C(x,y){:}(B(x){:}has(x,r))$. Conversely, if an agent x notifies an agent y that it does not hold a particular belief (say $B(x){:}has(x,r)$), then x may store this as a negative belief commitment to y, as follows: $C(x,y){:}(B(x){:}\neg has(x,r))$[4]. In both cases, the recipient agent (y) may also record the commitment in the same way in its own commitment store, but the commitment would be interpreted as a commitment of the sender (x) to y and would allow y to reason about x's beliefs. An example of a message that would be sent by x to retract a positive belief commitment is $retract(x, y, B(x){:}has(x,r))$, or $retract(x, y, B(x){:}\neg has(x,r))$ in the case of a negative belief commitment.

Commitment to Desire. As with beliefs, if an agent x notifies another agent y of a desire (say $D(x){:}has(x,r)$), then, depending on its policy, x may store this as a commitment to y, as follows: $C(x,y){:}(D(x){:}has(x,r))$. Similarly, if x notifies y of an absence of the desire, then x may store this as a commitment to y, as follows: $C(x,y) : (\neg D(x) : has(x,r))$[5]. As with belief commitments, the recipient agent (y) may record the notification as a commitment of the sender (x) to it, which would allow y to reason about x's desires. An example of a message that would

[4] We adopt a closed-world assumption for beliefs of the form $B(x){:}has(x,r)$. Hence, if an agent x does not believe it has a resource r, then x believes it does not have r.

[5] We assume a difference between an absence of desire and a desire of the form $D(x){:}\neg has(x,r)$. In the latter case, having r violates one of x's desires. This is not the case with an absence of desire, wherein x is indifferent to having r or not having r.

be sent by x to retract a desire commitment is $retract(x, y, D(x){:}has(x, r))$, or $retract(x, y, \neg D(x){:}has(x, r))$ in the case of an absence of desire.

Commitment to Dialogue. The dialogues considered in this paper, and explained in Section 3, are information-seeking and negotiation. If an agent (say x) sends a message to another agent that commences an instance of dialogue, then x may record this in its commitment store. An instantiation by x of an information-seeking dialogue for some information (say $Query$) from y would be recorded as $C(x, y) : dialogue(x, y, info, Query)$, and an instantiation of a negotiation dialogue for an exchange of resources (say Rx for Ry) would be recorded as $C(x, y) : dialogue(x, y, neg, Rx, Ry)$. Depending on the agent's internal policy, such commitments can serve a number of purposes, for example, prohibiting x from sending certain messages until other expected responses are received. Upon completion of the dialogue, x would be expected to remove its commitment to the dialogue. No explicit *retract* would be required, since it would be known to both agents that the dialogue is complete.

As with initiating a dialogue, refusing a certain dialogue request may add a dialogue commitment to the agent's commitment store, but the commitment here would be negative. For example, if an agent y refuses a certain request from x for some information (say $Query$), then y may store this in its commitment store as $C(y, x){:}\neg dialogue(x, y, info, Query)$, or if y refuses a certain request from x to exchange resources (say Rx for Ry) then y may store this in its commitment store as $C(y, x){:}\neg dialogue(x, y, neg, Rx, Ry)$. The agent receiving the refusal (x) may record y's commitment in its own commitment store, which could serve a number of purposes, for example, prohibiting x from attempting the same dialogue request again until y retracts the negative commitment. The owner of a negative dialogue commitment (say y) may retract its negative dialogue commitment by notifying the agents to whom it has the commitment with a message of the form $retract(y, x, \neg dialogue(x, y, info, Query))$ or $retract(y, x, \neg dialogue(x, y, neg, Rx, Ry))$.

Action Rules. Similar to the actions in [4], the actions available to an agent have associated with them preconditions, constraints and consequences. The preconditions and constraints limit what actions are legal for an agent to perform at a particular point in time: an agent can perform an action if at that point in time all of the preconditions of the action rule hold and none of the constraints hold. The consequences specify changes the agent must make to its internal state as a result of performing the action.[6]

An action rule is specified as a quadruple $< A, P, Q, X >$, where A is an action, P are the preconditions for performing the action, Q are the contraints and X are the consequences. The form that A takes is $Type[Message]$, as mentioned above. The $Message$ part of A may contain both grounded literals and variables[7]. P and Q are each a set of clauses. For P, the set of clauses is read as

[6] Agents are assumed to be autonomous, hence performing an action does not directly modify another agent's internal state.

[7] Variables occurring in an action rule are implicitly universally qualified from the outside.

a conjunction, such that all of the clauses must hold for the action to be applicable. For C, the set of clauses is read as a disjunction, such that any one of the clauses holding renders the action inapplicable. Each clause in P is a disjunction of terms and each clause in C is a conjunction of terms. The terms in the clauses of P and C are over the agent's beliefs, desires, messages and commitments, and may consist of both grounded literals and variables. X specifies a set of individual terms over the agent's beliefs, desires, messages and commitments that would be added and, if preceded by a $\sim$, removed as a result of performing A. The terms in X may consist of both grounded literals and variables. Examples of action rules can be found in Section 4.

Evaluation Mechanism. The task of the evaluation mechanism is to determine the actions that the agent may perform (i.e. the legal actions) by evaluating the preconditions and constraints of the action rules and finding instantiations of the variables. A unification procedure for matching terms similar to that of Prolog could be used. Indeed, the action rules, excluding the consequences, can be seen as logic program rules: the action (A) is interpreted as the conclusion (head) of the rule and the conjunction of precondition (P_i) and negated[8] constraint (C_i) clauses are interpreted as the conditions (body) of the rule, as follows: $A \leftarrow P_1 \wedge \ldots \wedge P_m \wedge \mathit{not}\ C_1 \wedge \ldots \wedge \mathit{not}\ C_n$. The disjunction of all the actions that are legal at a particular time is interpreted as the agent's intention at that point in time. After performing an action the agent re-evaluates its intention.

Preferences. Communication proceeds in the agent system on the basis of an alternation of turns between agents, wherein the agent whose turn it is can perform a number of actions before passing over control. However, only one action is performed at each time-point. The choice as to which action to perform is based on two factors: Firstly, which actions are legal for the agent at that point in time, determined by the action rules. Secondly, the agent's preferences, which are over the actions and specified by assigning a level of priority to each action: Actions of highest priority are assigned a level of 1, actions of second-highest priority are assigned a level of 2, and so on. The action chosen to be performed at a particular point in time is the legal action of lowest level (i.e. highest priority).

It may be possible for two different actions of the same priority level to be legal for the agent simultaneously. This is resolved in one of two ways: In the case of processing or responding to a received message, from the choice of legal actions of same priority, the action chosen is the one which corresponds to the oldest received message (i.e. a first-come first-serve basis).[9] In the case of sending a message that is not in response to a received message (e.g. initiating a dialogue), from the choice of legal actions of same priority, one is chosen at random. Although preferences over beliefs and desires should be a possibility for agents this will be ignored in this paper and left for future work.

[8] Negation as failure.

[9] This is possible since a queuing mechanism is assumed in the message store that is able to distinguish messages by the order in which they are received.

3 The Dialogues

Two types of dialogue are considered in this paper: information-seeking and negotiation. The dialogue protocols, as defined in this section, specify the permissible messages and the order in which the messages can be made. Note that both dialogues are defined as request-response message sequences between two agents. However, the dialogues differ in the purposes they serve and the messages that can be exchanged. Below, we present the protocols informally. A formal presentation, such as using finite state automata, is omitted for lack of space. In the process of sending and receiving the messages during the dialogues, the agents may adopt certain commitments depending on their internal policies. This is demonstrated in the next section.

Information-Seeking. An information-seeking dialogue as defined in this paper allows one agent to request and possibly obtain information from another. If an agent (say X) seeks the answer to some question from another agent (say Y), who is believed by the first to know the answer, it sends a message requesting the sought information, as follows: $inforeq(X, Y, Query)$. In this message $Query$ is the information sought, the meaning of which is assumed to be shared by both sender and receiver. As an example, given a $Query$ of the form $B(Y){:}has(Y, R)$, in the context of information-seeking the agents would interpret this as: "Does agent Y, the recipient of the $inforeq$ message, believe it has resource R?" This is a query seeking information about the recipient agent's beliefs. Other queries may be about the recipient agent's desires or commitments.

The agent receiving the $inforeq$ message (Y) is obliged by the dialogue protocol to complete the dialogue by either responding to the query or refusing to provide the requested information. In the case of responding to the query, a message of the form $inforesp(Y, X, Query, Response)$ is sent, where $Query$ is the information sought, as sent by X, and $Response$ is either *yes* or *no*. In the case of refusing to provide the requested information, a message of the form $inforef(Y, X, Query)$ is sent, where $Query$ is the information sought by X.

Negotiation. A negotiation dialogue as defined in this paper allows two agents to agree (or disagree) on an exchange of resources. If an agent (say X) seeks an exchange of resources with another agent (say Y), it sends an $offer$ message specifying the resource to be given (Rx) and received (Ry) in the exchange, as follows: $offer(X, Y, Rx, Ry)$.

The agent receiving the $offer$ message (Y) is obliged by the dialogue protocol to complete the dialogue by either accepting the offer or rejecting it. In the case of accepting, a message of the form $acceptoffer(Y, X, Rx, Ry)$ is sent, where Rx and Ry are the resources to be received and given away (respectively) by Y, as specified by X in its offer. In sending the acceptance, the resources are assumed to be exchanged and, accordingly, the agents are expected to update their beliefs to reflect this. In the case of rejecting the offer, a message of the form $rejectoffer(Y, X, Rx, Ry)$ is sent, where Rx and Ry are the resources as specified by X in its offer.

4 An Agent Policy for Bilateral Negotiation with Information-seeking

The dialogues protocols, as defined in the previous section, specify the allowed types of messages and how the messages are combined. It is the agent's policy (i.e. the action rules and preferences over the actions) and internal state (i.e. belief, desire, commitment and mesage stores), on the other hand, that determines the agent's behaviour, and how and when it fulfils the obligations imposed upon it by a dialogue. This is demonstrated in this section for an agent named x that can engage in the information-seeking and negotiation dialogues as both initiator and responder. For this agent the desires are fixed and attaining any one of the desired resources achieves the agent's overall goal. The agent is self-interested and hence offers or accepts an exchange of resources as long as the resource to be received is one that it desires. That is, unless it already has a desired resource, in which case it will not make any offers but will still accept an offer as long as a desired resource is received in exchange. The policy does not permit the agent to initiate more than one dialogue simultaneously. However, there is no limit to the number of dialogues it may be engaged in as responder. The agent does not send any requests for information once it has obtained a desired resource but will still continue to respond to information requests. A received message is put in the message store until such time that the agent can process it and/or respond. Any inconsistent commitments are retracted at the earliest oppurtunity.

The action rules are grouped in the following subsections by the dialogue to which they contribute and the role of the agent in that dialogue. The last subsection specifies preferences over the actions, which enables the agent to select a single action when there is a choice.

4.1 Information-Seeking Dialogue as Initiator

A= $send[inforeq(x, Y, (B(Y){:}has(Y, R)))]$
P= $\{D(x){:}has(x, R)\}$
Q= $\{C(Y, x){:}\neg dialogue(x, Y, info, (B(Y){:}has(Y, R))),$
$C(x, Y'){:}dialogue(x, Y', info, \_\ ),\ C(x, Y''){:}dialogue(x, Y'', neg, \_\ , \_\ ),$
$(B(x){:}has(x, R')) \wedge (D(x){:}has(x, R')),$
$C(Y, x){:}(B(Y){:}has(Y, R)),\ C(Y, x){:}(B(Y){:}\neg has(Y, R))\}$
X= $\{C(x, Y){:}dialogue(x, Y, info, (B(Y){:}has(Y, R)))\}$

Agent x sends an $inforeq$ message to an agent Y to determine if Y has a resource (R) that x desires. Agent x sends this message as long as x does not have a negative dialogue commitment from Y preventing it from doing so (first constraint), x is not committed to any information-seeking or negotiation dialogue as initiator (second and third constraints), x does not already have a desired resource (R') (fourth constraint) and x does not know whether Y has the resource R (fifth and sixth constraints). The single consequence of sending the $inforeq$ message is that x becomes committed to an instance of information-seeking dialogue.

A= $send[inforeq(x, Y, (D(Y){:}has(Y, R)))]$
P= $\{B(x){:}has(x, R),\ ((D(x){:}has(x, R')) \wedge (C(Y, x){:}(B(Y){:}has(Y, R'))))\}$
Q= $\{C(Y, x){:}\neg dialogue(x, Y, info, (D(Y){:}has(Y, R))),$
$C(x, Y'){:}dialogue(x, Y', info, \_),\ C(x, Y''){:}dialogue(x, Y'', neg, \_, \_),$
$(B(x){:}has(x, R'')) \wedge (D(x){:}has(x, R'')),$
$C(Y, x){:}(D(Y){:}has(Y, R)),\ C(Y, x){:}(\neg D(Y){:}has(Y, R))\}$
X= $\{C(x, Y){:}dialogue(x, Y, info, D(Y){:}has((Y, R)))\}$

This action rule is similar to the previous one, except here the information request is to determine whether Y has a desire for some resource (R) that x has. Note that x performs this action only if Y has a resource (R') desired by x (second precondition).

A= $process[inforesp(Y, x, Query, yes)]$	A= $process[inforesp(Y, x, Query, no)]$
P= $\{M(x){:}inforesp(Y, x, Query, yes),$ $C(x, Y){:}dialogue(x, Y, info, Query)\}$	P= $\{M(x){:}inforesp(Y, x, Query, no),$ $C(x, Y){:}dialogue(x, Y, info, Query)\}$
Q= $\{\}$	Q= $\{\}$
X= $\{\sim M(x){:}inforesp(Y, x, Query, yes),$ $\sim C(x, Y){:}dialogue(x, Y, info, Query),$ $C(Y, x){:}Query\}$	X= $\{\sim M(x){:}inforesp(Y, x, Query, no),$ $\sim C(x, Y){:}dialogue(x, Y, info, Query),$ $C(Y, x){:}\neg Query\}$

In processing a received $inforesp$ message (stored in M), x updates its message and commitment stores as shown in the consequences of the above two action rules[10]. The symbol $\sim$ denotes removal of a term.

A= $process[inforef(Y, x, Query)]$
P= $\{M(x){:}inforef(Y, x, Query),\ C(x, Y){:}dialogue(x, Y, info, Query)\}$
Q= $\{\}$
X= $\{\sim M(x){:}inforesp(Y, x, Query, no),\ \sim C(x, Y){:}dialogue(x, Y, info, Query),$
$C(Y, x){:}(\neg dialogue(x, Y, info, Query))\}$

In processing a refusal for some information, x updates its message and commitment stores as shown in the consequences of the above action rule.

4.2 Information-Seeking Dialogue as Responder

A= $send[inforesp(x, Y, (B(x){:}has(x, R)), yes)]$	A= $send[inforesp(x, Y, (B(x){:}has(x, R)), no)]$
P= $\{M(x){:}inforeq(Y, x, (B(x){:}has(x, R))),$ $B(x){:}has(x, R)\}$	P= $\{M(x){:}inforeq(Y, x, (B(x){:}has(x, R)))\}$
Q= $\{C(x, Y'){:}dialogue(x, Y', neg, R, \_),$ $C(x, Y''){:}dialogue(x, Y'', neg, \_, R)\}$	Q= $\{B(x){:}has(x, R),$ $C(x, Y'){:}dialogue(x, Y', neg, R, \_),$ $C(x, Y''){:}dialogue(x, Y'', neg, \_, R)\}$
X= $\{\sim M(x){:}inforeq(Y, x, (B(x){:}has(x, R))),$ $C(x, Y){:}(B(x){:}has(x, R))\}$	X= $\{\sim M(x){:}inforeq(Y, x, (B(x){:}has(x, R))),$ $C(x, Y){:}(B(x){:}\neg has(x, R))\}$

Agent x can send a positive or negative response for a query about one of its beliefs if it has or does not have (respectively) such a belief in its belief store whilst it is not committed to a negotiation dialogue that may alter this belief. In doing so, agent x adopts a commitment to agent Y to notify it of any changes.

A= $send[inforesp(x, Y, (D(x){:}has(x, R)), yes)]$	A= $send[inforesp(x, Y, (D(x){:}has(x, R)), no)]$
P= $\{M(x){:}inforeq(Y, x, (D(x){:}has(x, R))),$ $D(x){:}has(x, R)\}$	P= $\{M(x){:}inforeq(Y, x, (D(x){:}has(x, R)))\}$
Q= $\{\}$	Q= $\{D(x){:}has(x, R)\}$
X= $\{\sim M(x){:}inforeq(Y, x, (D(x){:}has(x, R))),$ $C(x, Y){:}(D(x){:}has(x, R))\}$	X= $\{\sim M(x){:}inforeq(Y, x, (D(x){:}has(x, R))),$ $C(x, Y){:}(\neg D(x){:}has(x, R))\}$

These two action rules are similar to the previous two, except here the information sought is about one of x's desires.

[10] In processing a negative response of a belief query, the negation operator would appear within the $Query$ term. The distinction is not made here for lack of space.

A= $send[inforef(x, Y, Query)]$
P= $\{M(x){:}inforeq(Y, x, Query)\}$
Q= $\{(Query = B(x){:}has(x, \_)),\ (Query = D(x){:}has(x, \_))\}$
X= $\{C(x, Y){:}\neg dialogue(Y, x, info, Query)\}$

Agent x refuses any information request that is not about one of its beliefs or desires.

4.3 Negotiation as Initiator

A= $send[offer(x, Y, Rx, Ry)]$
P= $\{B(x){:}has(x, Rx),\ D(x){:}has(x, Ry),$
$C(Y, x){:}(D(Y){:}has(Y, Rx),\ C(Y, x){:}(B(Y){:}has(Y, Ry))\}$
Q= $\{C(x, Y'){:}dialogue(x, Y', neg, Rx, \_),\ C(x, Y''){:}dialogue(x, Y'', neg, \_, Ry),$
$C(Y, x){:}\neg dialogue(x, Y, neg, Rx, Ry),\ (B(x){:}has(x, R)) \wedge (D(x){:}has(x, R))\}$
X= $\{C(x, Y){:}dialogue(x, Y, neg, Rx, Ry)\}$

Agent x sends an offer to give a resource Rx to an agent Y and to receive a resource Ry in return if x's internal state determines the exchange to be possible and individually rational for both agents, as defined in the preconditions above. The constraints for sending the offer are that x has offered Rx to another agent (Y'), x has requested Ry from another agent (Y''), x has a negative dialogue commitment determining the exchange to be unwanted by Y, or x already has a desired resource (R).

A= $process[rejectoffer(Y, x, Rx, Ry)]$	A= $process[acceptoffer(Y, x, Rx, Ry)]$
P= $\{M(x){:}rejectoffer(Y, x, Rx, Ry),$ $C(x, Y){:}dialogue(x, Y, neg, Rx, Ry)\}$	P= $\{M(x){:}acceptoffer(Y, x, Rx, Ry),$ $C(x, Y){:}dialogue(x, Y, neg, Rx, Ry)\}$
Q= $\{\}$	Q= $\{\}$
X= $\{\sim M(x){:}rejectoffer(Y, x, Rx, Ry),$ $\sim C(x, Y){:}dialogue(x, Y, neg, Rx, Ry),$ $C(Y, x){:}\neg dialogue(x, Y, neg, Rx, Ry)\}$	X= $\{\sim M(x){:}acceptoffer(Y, x, Rx, Ry),$ $\sim C(x, Y){:}dialogue(x, Y, neg, Rx, Ry),$ $\sim B(x){:}has(x, Rx),\ B(x){:}has(x, Ry)\}$

Upon processing rejection or acceptance of an earlier sent offer, x either forms a negative dialogue commitment or updates it beliefs as to which resources it has.

4.4 Negotiation as Responder

A= $send[acceptoffer(x, Y, Ry, Rx)]$
P= $\{M(x){:}offer(Y, x, Ry, Rx),\ B(x){:}has(x, Rx),\ D(x){:}has(x, Ry))\}$
Q= $\{C(x, Y'){:}dialogue(x, Y', neg, Rx, \_),\ C(x, Y''){:}dialogue(x, Y'', neg, \_, Ry)\}$
X= $\{\sim M(x){:}offer(Y, x, Ry, Rx),\ \sim B(x){:}has(x, Rx),\ B(x){:}has(x, Ry)\}$

Agent x sends acceptance of a received offer if it has a desire for the offered resource (Ry) and believes it has the requested resource (Rx). The resources to be exchanged must not be pending in some other negotiation that x has initiated. In sending the acceptance message, the exchange is agreed to and x updates its beliefs as to which resources it has. Subsequently, x may have to update some of its commitments. Note that x does not check whether Y actually has the offered resource (Ry) since it assumes agents to be honest.

A= $send[rejectoffer(x, Y, Ry, Rx)]$
P= $\{M(x){:}offer(Y, x, Ry, Rx)\}$
Q= $\{(B(x){:}has(x, Rx)) \wedge (D(x){:}has(x, Ry))\}$
X= $\{\sim M(x){:}offer(Y, x, Ry, Rx),\ C(x, Y){:}\neg dialogue(Y, x, neg, Ry, Rx)\}$

Agent x sends rejection of a received offer if the preconditions for accepting do not hold (i.e. x would not have a desired resource after the exchange).

4.5 Sending Retractions of Commitment

A= $send[retract(x, Y, (\neg dialogue(Y, x, neg, Ry, Rx)))]$
P= $\{C(x, Y){:}(\neg dialogue(Y, x, neg, Ry, Rx)),\ B(x){:}has(x, Rx),\ D(x){:}has(x, Ry)\}$
Q= $\{\}$
X= $\{\sim C(x, Y){:}(\neg dialogue(Y, x, neg, Ry, Rx))\}$

Agent x retracts negative commitment to a particular resource exchange if the conditions for accepting the exchange now hold.

A= $send[retract(x, Y, (B(x){:}\neg has(x, R)))]$	A= $send[retract(x, Y, (B(x){:}has(x, R)))]$
P= $\{C(x, Y){:}(B(x){:}\neg has(x, R)),\ B(x){:}has(x, R)\}$	P= $\{C(x, Y){:}(B(x){:}has(x, R))\}$
Q= $\{\}$	Q= $\{B(x){:}has(x, R)\}$
X= $\{\sim C(x, Y){:}(B(x){:}\neg has(x, R))\}$	X= $\{\sim C(x, Y){:}(B(x){:}has(x, R))\}$

Agent x retracts commitment to the belief that it does (not) have resource R, if it now believes it has (does not have, respectively) resource R.[11]

4.6 Processing Retractions of Commitment

A= $process[retract(Y, x, Commitment)]$	A= $process[retract(Y, x, Commitment)]$
P= $\{M(x){:}retract(Y, x, Commitment),$ $C(Y, x){:}(Commitment)\}$	P= $\{M(x){:}retract(Y, x, Commitment)\}$
Q= $\{\}$	Q= $\{C(Y, x){:}(Commitment)\}$
X= $\{\sim M(x){:}retract(Y, x, Commitment),$ $\sim C(Y, x){:}(Commitment)\}$	X= $\{\sim M(x){:}retract(Y, x, Commitment)\}$

Agent x retracts an agent Y's (positive or negative) commitment to belief, desire or dialogue if told by Y to do so and if x has such a commitment stored. Otherwise, the message received from Y is removed with no changes made to x's commitment store.

4.7 Preferences over Actions

A possible grouping of actions into priority levels, which would allow the agent to better engage in the information-seeking and negotiation dialogues, is as follows:

P_1: $process[Message]$	P_5: $send[inforesp(x, Y, Query, Response)]$
P_2: $send[retract(x, Y, Commitment)]$	$send[inforef(x, Y, Query)]$
P_3: $send[rejectoffer(x, Y, Ry, Rx)]$	P_6: $send[inforeq(x, Y, (D(Y){:}has(Y, R))]$
$send[acceptoffer(x, Y, Ry, Rx)]$	P_7: $send[inforeq(x, Y, (B(Y){:}has(Y, R))]$
P_4: $send[offer(x, Y, Rx, Ry)]$	

There are a couple of points to note. Firstly, all legal actions of processing a received message and hence updating the agent's internal state would be performed before any action of sending a message. This is so that messages are not sent based on beliefs and commitments that need to be changed or retracted. Secondly, responding to a message and completing a dialogue has priority over initiating a new dialogue. This is to avoid the opening of multiple dialogues unnecessarily and to avoid situations of deadlock in negotiation where each agent is waiting for the other to respond.

[11] Note that there are no actions for x to retract desire or positive dialogue commitments. This is because x's desires are fixed (i.e. do not change) and a positive dialogue commitment is automatically relinquished as soon as the dialogue is complete.

5 Example

In this section we present an example of the information-seeking and negotiation dialogues that takes place in a system of three agents (x, y, z) and three resources (*apple*, *banana*, *pear*). Each agent has a policy as defined in the previous section, begins with empty commitment and message stores, and starts with beliefs and desires as follows:

$D(x){:}has(x, banana)$ $B(x){:}has(x, apple)$	$D(y){:}has(y, pear)$ $B(y){:}has(y, banana)$	$D(z){:}has(z, apple)$ $D(z){:}has(z, banana)$ $B(z){:}has(z, pear)$

It is assumed that each agent knows which agents and resources make up the system. The communication between the agents proceeds in a turn-based manner and the agent whose turn it is passes over control when it has no legal actions available to it. The details of turn-taking and the control-level dialogues are ignored in this paper and it is assumed that agents are aware when it is their turn in the communication. At each point in time only one action is selected and it is assumed that the chosen action is immediately executed. Based on this set-up, the dialogues between agents x, y, z (taking turns in that order) begin, as shown below. In this particular example all of the agents manage to obtain a desired resource. Note, for reasons of space, the internal actions (i.e. *process*) are not shown and only the messages exchanged are shown.

(1) $inforeq(x, y, (B(y){:}has(y, banana)))$	(11) $offer(y, z, banana, pear)$
(2) $inforesp(y, x, (B(y){:}has(y, banana)), yes)$ $inforeq(y, x, (B(x){:}has(x, pear))$	(12) $acceptoffer(z, y, banana, pear)$ $retract(z, x, (B(z){:}\neg has(z, banana)))$
	$retract(z, y, (B(z){:}has(z, pear)))$
(3) $inforeq(z, x, (B(x){:}has(x, apple))$	$inforesp(z, y, (D(z){:}has(z, banana)), yes)$
(4) $inforesp(x, y, (B(x){:}has(x, pear)), no)$	
$inforesp(x, z, (B(x){:}has(x, apple)), yes)$	(13) $inforeq(x, z, (B(z){:}has(z, banana)))$
$inforeq(x, y, (D(y){:}has(y, apple)))$	(14) $retract(y, x, (B(y){:}has(y, banana)))$
(5) $inforesp(y, x, (D(y){:}has(y, apple)), no)$	(15) $inforesp(z, x, (B(z){:}has(z, banana)), yes)$
$inforeq(y, z, (B(z){:}has(z, pear)))$	(16) $inforeq(x, z, (D(z){:}has(z, apple)))$
(6) $inforesp(z, y, (B(z){:}has(z, pear)), yes)$	(17) (no actions for y to perform)
$inforeq(z, x, (D(x){:}has(x, pear)))$	(18) $inforesp(z, x, (D(z){:}has(z, apple)), yes)$
(7) $inforesp(x, z, (D(x){:}has(x, pear)), no)$	(19) $offer(x, z, apple, banana)$
$inforeq(x, z, (B(z){:}has(z, banana)))$	(20) (no actions for y to perform)
(8) $inforeq(y, z, (D(z){:}has(z, banana)))$	(21) $acceptoffer(z, x, apple, banana)$
(9) $inforesp(z, x, (B(z){:}has(z, banana)), no)$	$retract(z, x, B(z){:}has(z, banana))$
$inforesp(z, y, (D(z){:}has(z, banana)), yes)$	(22) $retract(x, y, (B(x){:}\neg has(x, pear)))$
$inforeq(z, y, (B(y){:}has(y, apple))$	$retract(x, z, (B(x){:}has(x, apple)))$
(10) (no *send* actions for x to perform)	

6 Results and Discussion

In this section we discuss various properties desired of agent policies and dialogues that seek to provide a solution to the resource re-allocation problem, focusing on the self-interested agent policy defined in this paper.

6.1 Agent Policy Properties

The agent policy properties that we consider in this subsection are as found in [2, 5]. We discuss informally to what extent these properties are satisfied by the agent policy of Section 4.

Truthfulness. *Does the agent communicate non-deceptively?* This is the case for agents based on the defined policy since agents communicate in accordance with their true internal states (i.e. they communicate their true beliefs and desires) and retract inconsistent commitments at the earliest oppurtunity. Further, a resource is only offered for exchange if the agent has that resource and has not offered it to any other agent, hence, the agent is guaranteed to have the offered resource at the time of acceptance.

Weak Conformance. *Does the agent never utter an illegal dialogue move wrt the protocols?* This is evident for the defined agent policy since the only communicative actions available are those that are allowed by the protocols. Also, the action rules only allow the agent to send a response message ($inforesp$ / $inforef$, $acceptoffer$ / $rejectoffer$) if the corresponding dialogue initiation message ($inforeq$, $offer$) has been received.

Exhaustiveness. *Will the agent utter at least one response message for any dialogue initiation message ($inforeq$, $offer$) it receives?* For both types of dialogue, whilst the agent is not waiting for some other agent to respond to an $offer$ message that it has sent, the conditions of the actions rules for responding to a message cover every possible state that the agent may be in. Hence, at least one of the possible dialogue follow-ups is guaranteed to be part of the agent's intention as long as all agents being waited upon are guaranteed to respond with an expected follow-up. This has the potential of deadlock, where each agent is waiting for the other to respond. However, for a system of agents based on the policy defined in this paper, where agents take turns in the communication and responding to a dialogue request takes precedence over initiating a new dialogue, situations of deadlock are avoided and exhaustiveness can be guaranteed.

Determinism. *Will no more than one of the possible dialogue responses be generated/selected by the agent?* The conditions of the action rules for responding to a particular $inforeq$ or $offer$ message are non-overlapping, i.e., the permitted dialogue responses have conditions that disallow more than one to hold for the agent simultaneously. Hence, at most one of the possible dialogue follow-ups will be part of the agent's intention.

Robustness. *As well as being exhaustive, for any illegal message received (wrt the protocols), will the agent recognise it as illegal and utter a special dialogue message notifying the other agent of this?* Although the defined agent policy is not robustly conformant to the protocols, it is not difficult to add further action rules to recognise and deal with received messages that are illegal wrt the protocols. However, as discussed in [2], if agents are known to be weakly conformant, it is theoretically unnecessary to deal with robust conformance (since no agent will ever utter an illegal move). Also, the additional notification messages may burden communication channels unnecessarily and simply ignoring illegal messages would be a better strategy.

6.2 Dialogue Properties

The dialogue properties that we consider in this subsection are as found in [5]. We discuss informally whether these properties are satisfied for a system of agents based on the policy of Section 4[12]. The latter two properties (soundness and completeness) concern identifying solutions to the resource re-allocation (rrp) problem. Solutions are defined as follows: A solution of the rrp for an agent x is a distribution of resources such that x has the resources it requires[13]. A solution of the rrp for an agent system is a distribution of resources such that each agent has the resources it requires.

Termination. *Are inidividual dialogues and the overall communication guaranteed to finitely come to a halt?* Firstly, since exhaustiveness and determinism can be guaranteed for a system of agents based on the policy defined in this paper, as discussed in the previous subsection, agents are guaranteed to respond to messages with one and only response message. Hence, individual dialogues are guaranteed to terminate. Secondly, since agents adopt commitments that serve to prohibit them from repeating dialogues unnecessarily, the sequence of dialogues between the agents is guaranteed to converge such that overall communication terminates, regardless of the initial distribution of resources, turn-ordering and regardless of whether a solution exists.

Soundness. *If some agent x ends communication in a successful state (i.e. believing it has a desired resource), is the rrp of x solved? If every agent in the agent system ends communication in a successful state, is the rrp of the agent system solved?* As for the first issue, since agents begin with correct beliefs as to which resources they have, are truthful so any change of beliefs resulting from negotiation reflects correctly the resources that the agents have and because an agent retains at least one of its desired resources once obtained, the agent ends communication in a successful state if either it began communication believing it has a desired resource or at some point it agreed to an offer to receive a desired resource. Therefore, an agent x ending communication in a successful state does indeed have a desired resource and hence the rrp of x is indeed solved. Following on from this, if every agent ends communication in a successful state, then every agent has a desired resource and hence the rrp of the agent system is solved.

Completeness. *If a solution to the rrp exists for an agent system, is every agent guaranteed to end communication in a successful state, producing such a solution?* Since agents are truthful, all inconsistent commitments are retracted and because an agent will ask each other agent in the system one by one about their beliefs/desires until either a resource exchange is found or all agents have been asked, any possible exchange of resources between two agents (because each has a resource desired by the other) is guaranteed to be found and agreed to by the agents, producing a solution to the rrp for the two agents. However, such exchanges, in which both agents obtain a desired resource, alone are not enough to guarantee a solution of the rrp for the agent system. Consider the following two examples, where x, y, z are agents and a, b, c are resources:

[12] Note that it is not strictly necessary for all agents in an agent system to be based on the same policy.

[13] In the case of the agent policy we consider in this paper, a solution of the rrp for an agent x is a distribution of resources such that x has one of its desired resources.

1. x has a and desires c. y has b and c, and desires b. Here a solution for the agent system exists wherein x has c and y has b. However, were x to offer y an exchange of a for c, y would refuse since it does not desire a.
2. x has a and desires b. y has b and desires c. z has c and desires a. This cyclic situation poses an interesting challenge: No exchanges will be made since no two agents have the resource desired by the other, even though an overall solution exists.

From these two examples it is clear that a system of agents based on the self-interested policy of Section 4 is not guaranteed to find a solution, even though one may exist. These two examples demonstrate the need for agents to convince one another to agree to exchanges even when there is no direct benefit for the agent. The obvious tool to allow for this is persuasion/argumentation, as will be considered in future work.

7 Conclusion and Future Work

We have presented an agent model that allows private generative agent policies to be defined such that agents can cooperatively engage in dialogues of various kinds whilst attempting to fulfil their own individual desires. The two types of dialogue considered in this paper for resource re-allocation are information-seeking and negotiation. Basic request-response protocols for the two types of dialogue have been defined. In order to demonstrate the potential of the framework, an example policy has been defined that allows for agents to engage in the information-seeking and negotiation dialogues. Some properties of the policy and resulting dialogues have been discussed, and the need for persuasion (argumentation) in order to better the chances of reaching a solution has been touched upon. Future work will concentrate on guaranteeing the property of completeness for self-interested agents that are communicating in the context of the resource re-allocation problem by including the capability of persuasion.

References

1. Elizabeth Black and Anthony Hunter. A generative inquiry dialogue system. In *AAMAS 2007*, 2007.
2. Ulrich Endriss, Nicholas Maudet, Fariba Sadri, and Francesca Toni. Protocol conformance for logic-based agents. In *International Joint Conferences on Artificial Intelligence*, 2003.
3. Peter McBurney and Simon Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 2002.
4. Sonia V. Rueda, Alejandro J. Garcia, and Guillermo R. Simari. Argument-based negotiation among bdi agents. *Journal of Computer Science and Technology*, 2(7), october 2002.
5. Fariba Sadri, Francesca Toni, and Paolo Torroni. Dialogues for negotiation: Agent varieties and dialogue sequences. In *ATAL 2001*, pages 405–421. Springer-Verlag, 2002.
6. Douglas N. Walton and Erik C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, 1995.