A framework for argumentation-based negotiation

Carles Sierra[‡⋆], Nick R. Jennings[‡], Pablo Noriega[†⋆⋆], Simon Parsons[‡]

[‡] Department of Electronic Engineering,
Queen Mary and Westfield College,
University of London, London E1 4NS, UK.
{C.A.Sierra, N.R.Jennings, S.D.Parsons}@qmw.ac.uk

[†] Artificial Intelligence Research Institute, IIIA.
Spanish Scientific Research Council, CSIC.
Campus UAB, 08193 Bellaterra, Barcelona, Spain.
{sierra, pablo}@iiia.csic.es

Abstract. Many autonomous agents operate in domains in which the co-operation of their fellow agents cannot be guaranteed. In such domains negotiation is essential to persuade others of the value of co-operation. This paper describes a general framework for negotiation in which agents exchange proposals backed by arguments which summarise the reasons why the proposals should be accepted. The argumentation is persuasive because the exchanges are able to alter the mental state of the agents involved. The framework is inspired by our work in the domain of business process management and is explained using examples from that domain.

Keywords: Automated negotiation, Argumentation, Persuasion.

1 Introduction

Negotiation is a key form of interaction in systems composed of multiple autonomous agents. In such environments, agents often have no inherent control over one another and so the only way they can influence one another's behaviour is by persuasion. In some cases, the persuadee may require little or no convincing to act in the way desired by the persuader, for example because the proposed course of action is consistent with their plans. However, in other cases, the persuadee may be unwilling to accept the proposal initially and must be persuaded to change its beliefs, goals or preferences so that the proposal, or some variant thereof, is accepted. In either case, the minimum requirement for negotiation is for the agents to be able to make proposals to one another. These proposals can then either be accepted or rejected as is the case in the contract net protocol [17], for instance. Another level of sophistication occurs when recipients do not just have the choice of accepting or rejecting proposals, but have the option of making

[⋆] On sabbatical leave from IIIA[†] thanks to a Spanish MEC grant PR95-313. Research partially supported by the Spanish CICYT project SMASH, TIC96-1038-C04001.

[⋆⋆] On leave from *Laboratorio Nacional de Informática Avanzada*—LANIA. Rébsamen, 80; Xalapa, Veracruz, Mexico. Enjoying a Mexican CONACYT grant [69068-7245].

counter offers to alter aspects of the proposal which are unsatisfactory [16]. An even more elaborate form of negotiation—argumentation-based—is that in which parties are able to send justifications or arguments along with (counter) proposals indicating why they should be accepted [11, 13, 18]. Arguments such as: "this is my final offer, take it or leave it", "last time this job cost $5, I'm not going to pay $10 now", and "the job will take longer than usual because one of the workers is off sick" may be necessary to change the persuadee's goals or preferences.

This paper deals with argumentation-based negotiation. Because this is a large research topic [9, 19] we limit our scope to argumentation between computational agents where a persuader tries to convince a persuadee to undertake a particular problem solving task (service) on its behalf. We outline the components of a formal model for the process of argumentation-based negotiation which can ultimately be used to build negotiating agents for real world applications. While we draw on our previous work in this area, in this paper we shift our attention from the mechanisms for generating counter proposals [16] and those for generating and interpreting arguments [13] to the social aspects of the negotiation. Moreover, we take advantage of the work on Dialogical Frameworks introduced in [12] to define the static aspects of the negotiation process: shared ontology, social relations, communication language and protocol. We define a minimal notion of the *state* of an agent which captures the evolutionary character of negotiation—enabling the resulting model to recognise different types of arguments that agents can make in support of their proposals. Finally, we indicate how these arguments can be generated and interpreted by agents.

In the paper we discuss three types of illocutions: (i) *threats*—failure to accept this proposal means something negative will happen to the agent; (ii) *rewards*—acceptance of this proposal means something positive will happen to the agent; and (iii) *appeals*—the agent should prefer this option over that alternative for this reason. We realise these are a subset of the illocutions that are involved in persuasive negotiation (see [9] for a list based on psychological research), but our emphasis is in providing an overarching framework in which the key components of argumentation can be described, rather than providing an exhaustive formalisation of all the argument types which can be found in the literature. We illustrate these constructs through a running example introduced in the following section. The main contribution of this work is, therefore, to provide a formal framework in which agents can undertake persuasive negotiation to change each other's beliefs and preferences using an expressive communication language. Moreover, the framework is neutral with respect to the agent's internal architecture and imposes few constraints on its formal resources.

2 Argumentation in Business Process Management

This section describes the scenario which will be used to illustrate the principles and concepts of our model of argumentation. The scenario is motivated by work in the ADEPT project [8] which has developed negotiating agents for business process management applications. In particular, we consider a multi-agent system for managing a British Telecom (BT) business process—namely, providing a quotation for designing a network which offers particular services to a customer (Figure 1). The overall process receives a customer service request as its input and generates as its output a quote specifying how much it would cost to build a network to realise that service. Here we consider a subset of the agents involved in this activity: the customer service division (CSD) agent, the design division (DD) agent, the surveyor department (SD) agent, and the various agents who provide the out-sourced service of vetting customers (VC agents). A full account of all the agents and their negotiations is given in [16].

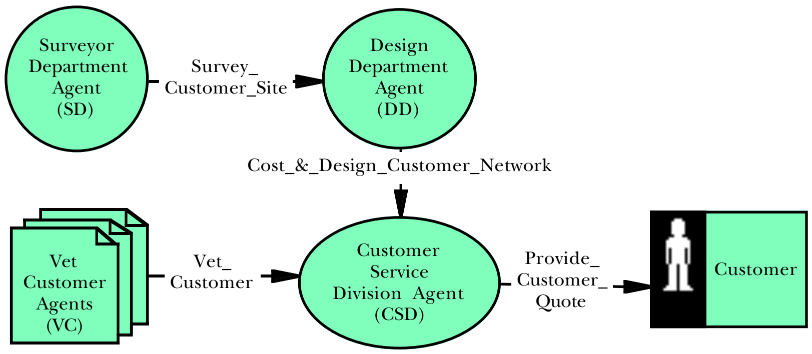
Surveyor Department Agent (SD)
Survey_ Customer_Site
Design Department Agent (DD)
Cost_&_Design_Customer_Network
Vet Customer Agents (VC)
Vet_ Customer
Customer Service Division Agent (CSD)
Provide_ Customer_ Quote
Customer

Fig. 1. Agent system for BT's "$Provide\_Customer\_Quote$" business process. The direction of the arrow indicates who provides the service labelling the arrow to whom.

The first stages of the Provide_Customer_Quote service involve the CSD agent capturing basic information about the customer and vetting the customer in terms of their credit worthiness. The latter service is performed by one of the VC agents and negotiation is used to determine which one is selected. If the customer fails the vetting procedure, then the quote process terminates. Assuming the customer is satisfactory, the CSD agent maps their requirements against a service portfolio. If the requirements can be met by a standard off-the-shelf portfolio item then an immediate quote can be offered based on previous examples. In the case of bespoke services the process is more complex. The CSD agent negotiates with the DD agent for the service of costing and designing the desired network service. To prepare a network design it is usually necessary to have a detailed plan of the existing equipment at the customer's premises. Sometimes such plans might not exist and sometimes they may be out of date. In either case, the DD agent determines whether the customer site(s) should be surveyed. If such a survey is warranted, the DD agent negotiates with the SD agent for the Survey_Customer_Site service. This negotiation differs from the others present in this scenario in that the two agents are part of the same department. Moreover, the DD agent has a degree of authority over SD. Agent negotiation is still required to set the timings of the service, but the SD agent cannot simply refuse to perform the service. On completion of the network design and costing, the DD agent informs the CSD agent which informs the customer of the service quote. The business process then terminates.

The precise nature of the argumentation which can occur in the aforementioned negotiations is determined by three main factors: (i) the negotiation arity—pairwise (1 to 1) negotiations (e.g. the CSD and DD agents for the design network service) differ from 1 to many negotiations (e.g. the CSD and VC agents for the Vet_Customer

Type	Id	Parties	Content	Comments
Threaten	1	CSD-VCs	Match the offer I have from another VC, otherwise I'll break off this negotiation.	Threaten to terminate current negotiation thread.
	2	CSD-VCs	Make sure you get back to me in the specified time period or I won't involve you in future rounds of bidding.	Threaten to terminate all future negotiation threads.
	3	DD-SD	If you cannot complete the service sooner, I'll inform your boss that we missed the deadline because of you.	Threaten to inform outside party of (perceived) poor performance.
Reward	4	CSD-DD	If you produce this design by this time we'll be able to get the quote to our major customer ahead of time.	Indicate positive effect of performing action by specified time.
	5	CSD-VCs	If you vet this customer by this time, I'll make sure you're involved in subsequent rounds of bidding.	Promise future involvement for accepting current proposal.
Appeal	6	CSD-VCs	Last time you vetted this customer, it took this length of time and cost this much.	Appeal to precedent.
	7	CSD-DD	You must complete this design within 48 hours because company policy says customers must be responded to within this time frame.	Appeal to (company's) prevailing practice.
	8	VC-CSD	This customer may be in financial trouble, therefore more time is needed to carry out a higher quality vetting.	Appeal to (CSD's) self interest.
	9	DD-CSD	The design will take longer than normal because one of our surveyors is on holiday this week.	Revealing new information.
	10	SD-DD	Customer has many premises and they all need to be surveyed, thus this service will take longer than normal.	Revealing new information.

Fig. 2. Sample arguments in the BT application.

service); (ii) the power relations [2] between the negotiators—most negotiations are peer-to-peer, but the DD and SD negotiation over the Survey_Customer_Site service is an example of boss-to-subordinate negotiation; and (iii) the organisational relationship of the negotiators—some negotiations are between agents of the same organisation (e.g. the CSD, DD and SD agents), while others are between agents of different organisations (e.g. the CSD and VC agents). Our experience in the domain shows that the argumentation between agents can be captured by the three types of argument mentioned in the Introduction—threats, rewards and appeals. Some examples of such arguments are given in Figure 2.

3 Negotiation model

Our model describes the process of a single encounter negotiation between multiple agents over a deal. Deals are always between two agents, though an agent may be engaged simultaneously in negotiation with many agents for a given deal. Negotiation is achieved through the exchange of illocutions in a shared communication language CL. The actual exchange of illocutions is driven by the participating agents' *individual* needs and goals—something that will not be part of this negotiation model. Nevertheless, this exchange is subject to some *minimal shared conventions* on the intended usage of the illocutions in CL, and a simple negotiation protocol. These conventions relate to:

1. The elements that are relevant for the negotiation of a deal—in the form of *issues* and *values* that may evolve as negotiation proceeds.

2. The rationality of the participating agents—in terms of some form of preference relationships or utility functions which enable the agents to evaluate and compare different proposals.
3. The deliberation capability of the participating agents—in the form of an internal *state* in which the agent may register the history of the negotiation as well as the evolution of its own theoretical elements on which its decisions are founded.
4. The minimal shared meaning of the acceptable illocutions—this is captured in the way that a *received* illocution should be interpreted when heard by an agent, and by making explicit the conditions that enable an agent to use (or 'generate') a given illocution at a given time.

A minimal set of concepts which are necessary to represent the static components in automated negotiation are presented in Section 3.1, and the dynamic components—the concepts of a negotiation thread and a negotiation state—are introduced in Section 3.2. Social aspects that are relevant for persuasive arguments are dealt with in Section 3.3, and the process of interpreting and generating illocutions is illustrated in Section 3.4.

3.1 A Basic Negotiation Ontology

Negotiation requires communication between the agents and, for it to be unambiguous, each agent must have a unique identifier. We denote the set of identifiers of the agents involved in a negotiation as $Agents$[3]. The agents involved in a negotiation will have a variety of social relationships with one another. These relationships have an important impact upon the persuasion and argumentation process. For instance, prestigious speakers have a large persuasive impact and peers can be persuaded more easily than non-peers [9]. To model this characteristic, we assume that a general and shared social relation is defined between the agents. This relation can be modelled as a binary function over a set of social roles, denoted as $Roles$. In the BT scenario, for example, $Roles$ would be: $\{Customer, Contractor, Boss, Peer\}$. Finally, we assume that agents, when negotiating, interchange illocutions in a common communication language CL defined over a set of illocutionary particles whose propositional content is expressed in a shared logical language L[4]. The precise nature of L is unimportant in our model (e.g. it could be a propositional language or a modal language), however it must contain at least the following:

1. *Variables*. To represent the issues under negotiation. They have to be variables because issues need to be bound to different values during negotiation.
2. *Constants*. To represent values for the issues under negotiation. A special constant '?' is needed to represent the absence of value, and allow for underdefined proposals between agents. (Note this constant does not mean "don't care".)

[3] In practice, this set may change dynamically (e.g. new vetting companies may be created and old ones may disappear). However, since this process can be seen as independent from the negotiation process, our model is presented with respect to a fixed set.

[4] In practice, agents often have heterogeneous information models and so need to use one of the variety of techniques for allowing them to interoperate [5, 7]. However, in this work we adopt the simplest solution and assume a common language.

3. *Equality.* To specify the value of an issue under negotiation.
4. *Conjunction.* To define complex sentences.

All of these features are necessary to express the kinds of sentences involved in the negotiation proposals discussed in this paper. An example of such a sentence is:

$$(Price = \pounds 10) \wedge (Quality = High) \wedge (Penalty = ?)$$

where '$Price$', '$Quality$', and '$Penalty$' are the issues under negotiation and so are represented as variables; '$\pounds 10$', '$High$', and '?' are values for those issues and so are constants; '$=$' denotes equality; and '$\wedge$' denotes conjunction. However, the language defined so far is not expressive enough to describe everything that is involved in a negotiation. In particular, to 'reason' and 'argue' about offers it is necessary at the very least to have some way of expressing preferences between offers. Offers are formulae in L, hence the most obvious way of representing preferences between formulae would be as a second-order relation in L. However, this would mean that L would be a higher-order logic, with the associated computational problems of such logics [6]. As a result we prefer to express preferences as a meta-language ML with the following minimum requirements:

1. *Quoting functions.* To represent formulae in L as terms in ML.
2. *A preference meta-predicate.* To express preferences between formulae in L.

For example, given the sentences $Price = \pounds 10$, and $Price = \pounds 20$ in L, we can express a preference for the first over the second as:

$$Pref(equal(\lceil Price \rceil, \lceil \pounds 10 \rceil), equal(\lceil Price \rceil, \lceil \pounds 20 \rceil))$$

where '$equal$' is the quoting in ML of the predicate '$=$' in L, and '$Pref$' represents the preference meta-predicate. In the remainder of the paper, instead of writing $equal(\lceil Price \rceil, \lceil \pounds 10 \rceil)$ the more compact representation $\lceil Price = \pounds 10 \rceil$ is used.

The common communication language, CL, accounts for the set of illocutionary particles necessary to model the set of illocutionary acts we study in this paper. The acts can be divided into two sets, I_{nego} corresponding to negotiation particles (those used to make offers and counter offers) and I_{pers} corresponding to persuasive particles (those used in argumentation). $I_{nego} = \{\texttt{offer}, \texttt{request}, \texttt{accept}, \texttt{reject}, \texttt{withdraw}\}$, $I_{pers} = \{\texttt{appeal}, \texttt{threaten}, \texttt{reward}\}$. Other illocutions could conceivably be brought into CL but the present set is sufficient for our purposes.

The negotiation dialogue between two agents consists of a sequence of offers and counter offers containing values for the issues. These offers and counteroffers can be just conjunctions of '$issue = value$' pairs (`offer`) or can be accompanied by persuasive arguments (`threaten`, `reward`, `appeal`). 'Persuasion' is a general term covering the different illocutionary acts by which agents try to change other agent's beliefs and goals. The selection of three persuasive particles in the set I_{pers} is the result of an analysis of the domain, as explained in Section 2, as well as of the persuasion literature [9, 18]. `appeal` is a particle with a broad meaning, since there are many different types of appeal. For example, an agent can appeal to authority, to prevailing practice or to self-interest [18]. The structure of the illocutionary act is $\texttt{appeal}(a, b, \xi, [not]\varphi, t)$, where φ is the argument—a formula in L or in ML, or an illocution in CL—that agent a communicates to b in support of a formula ξ (which may be a formula either in L or ML). All types of appeal adhere to this structure. The differing nature of the appeal is achieved by varying the φ in L or ML or by varying $[not]\varphi$ in CL—*not* φ is understood as the fact that action φ does not take place. `threaten` and `reward` are simpler because they have a narrower range of interpretations. Their structure, $\texttt{threaten}(a, b, [not]\psi_1, [not]\psi_2, t)$ and $\texttt{reward}(a, b, [not]\psi_1, [not]\psi_2, t)$ is recursive since formulae ψ_1 and ψ_2 again may be illocutions in CL. This recursive definition allows for a rich set of possible (illocutionary) actions supporting the persuasion. For instance, agent DD can threaten agent SD that it will inform SD's boss about SD's incompetence if SD does not accept a particular deal:

$$\begin{aligned}&\texttt{threaten}(DD, SD, not\ \texttt{accept}(SD, DD, time = 24h, t_2),\\ &\quad\texttt{appeal}(DD, Boss\_of\_SD, SD = incompetent,\\ &\qquad not\ \texttt{accept}(SD, DD, time = 24h, t_2), t_3), t_1)\end{aligned}$$

Having introduced all the components, we can now describe our dialogical framework for persuasive negotiation.

Definition 1. A *Dialogical Framework* is a tuple $DF = \langle Agents, Roles, R, L, ML, CL, Time\rangle$, where

1. $Agents$ is a set of agent identifiers.
2. $Roles$ is a set of role identifiers.
3. $R : Agents \times Agents \to Roles$, assigns a social role to each pair of agents. Social relations can therefore be viewed as a labelled graph.
4. L is a logical language[5] satisfying the requirements mentioned above. $Deals(L)$ denotes the set of all possible conjunctive formulae in L over equalities between issues and values, i.e. $x_1 = v_1 \wedge \ldots \wedge x_n = v_n$. $Deals_{?\text{-}free}(L) \subset Deals(L)$ excludes '?' as an acceptable value in a deal.
5. ML is a metalanguage over L satisfying the requirements mentioned above.
6. CL is the language for communication between agents. Given $a, b \in Agents$ and $t \in Time$ it is defined as:
   (a) if $\delta \in Deals(L)$ then $\texttt{request}(a, b, \delta, t) \in CL$.
   (b) if $\delta \in Deals_{?\text{-}free}(L)$ then $\texttt{offer}(a, b, \delta, t)$, $\texttt{accept}(a, b, \delta, t)$, $\texttt{reject}(a, b, \delta, t) \in CL$.
   (c) $\texttt{withdraw}(a, b, t) \in CL$.
   (d) if $\psi_1, \psi_2 \in CL$, $\xi \in L \cup ML$, and $\varphi \in L \cup ML \cup CL$ then $\texttt{threaten}(a, b, [not]\psi_1, [not]\psi_2, t)$, $\texttt{reward}(a, b, [not]\psi_1, [not]\psi_2, t)$, $\texttt{appeal}(a, b, \xi, [not]\varphi, t) \in CL$.
7. $Time$ is a discrete totally ordered set of instants.

Note that the time stamp, which appears as the last argument in all illocutions, will be omited when there is no ambiguity.

Agents can use the illocutions in CL according to the following negotiation protocol (see Figure 3):

[5] In keeping with the spirit of specifying a framework which is neutral with respect to the agent architecture, we do not commit to any specific formal language but note that L could be as simple as a propositional language or as elaborate as a multi-modal BDI logic [10, 14].

1. A negotiation always starts with a *deal proposal*, i.e. an `offer` or `request`. In `request` illocutions the special constant '?' may appear. This is thought of as a petition to an agent to make a detailed proposal by filling the '?'s with defined values.
2. This is followed by an exchange of possibly many counter proposals (that agents may `reject`) and many persuasive illocutions.
3. Finally, a *closing* illocution is uttered, i.e. an `accept` or `withdraw`.

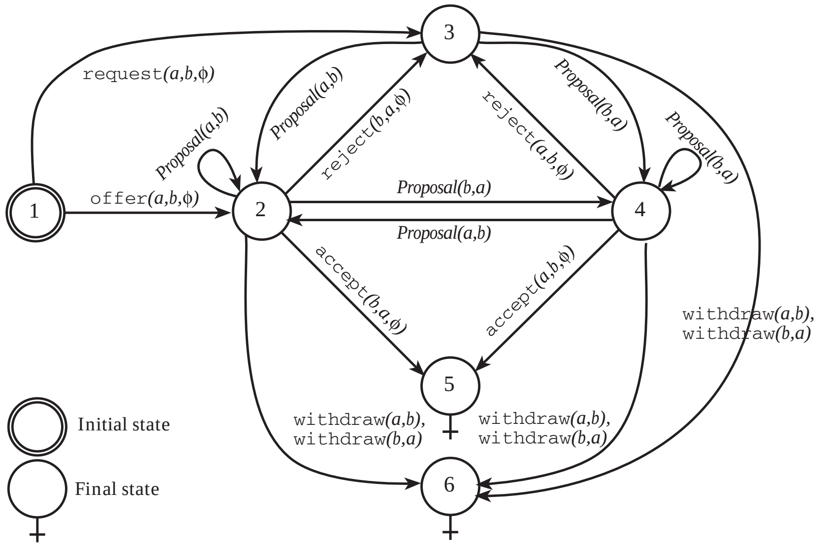

Fig. 3. Negotiation protocol. In $\texttt{accept}(x, y, \varphi)$ and $\texttt{reject}(x, y, \varphi)$ illocutions φ always refer to the last proposal. $Proposal(x, y)$ stands for any illocution constructed with any of the following particles: `offer`, `threaten`, `reward`, `appeal`, and between agents x and y. We omit the time stamp in the illocutions.

3.2 Negotiating agents

The Dialogical Framework described in the previous section represents the static components of the negotiation model—those that are fixed for all negotiations. This section presents the dynamic elements—those that change as a particular negotiation proceeds. Although our model aims to be as neutral as possible about the agent architecture, in order to capture essential aspects of persuasion it is necessary to assume that the agents have memory and are deliberative. Memory is expressed by means of an evolving *negotiation state* which, in turn, requires the notion of a *negotiation thread* [12] to capture the history of the negotiation dialogue between a pair of agents.

Definition 2. A **Negotiation Thread** between agents $a, b \in Agents$, at time $t \in Time$, noted $\vartheta^t_{a\leftrightarrow b}$, is a finite sequence (ordered on $Time$) of the form $\langle x^{t_j}_{d_i \to e_i} : t_j \leq t \rangle$ where:

1. $x^{t_j}_{d_i \rightarrow e_i} \in CL$,
2. $d_i, e_i \in \{a, b\}$, the thread contains only illocutions between agents a and b,
3. $d_i \neq e_i$, the illocutions are *between* agents, and
4. if $t_k < t_l$ then $issues(x^{t_k}_{d_i \rightarrow e_i}) \subseteq issues(x^{t_l}_{d_j \rightarrow e_j})$, where $issues(x)$ represents the set of issues mentioned in illocution x. That is, we assume monotonicity over the set of issues under negotiation, so that once an issue has been brought into the negotiation, it is never supressed. We will use ellipsis whenever useful to make more compact expressions.

We denote the last illocution in a thread as $\check{\vartheta}$. We say a negotiation thread ϑ is **active** if $\check{\vartheta}$ is not an `accept` or `withdraw` illocution.

In an extension to our previous work [16], we want to capture the idea that new issues may arise during the negotiation process. This is necessary because we consider that one of the main ways in which an agent may persuade another about the desirability of a particular proposal is to introduce new issues that have hitherto not featured in the thread. This means that we need an explicit representation of the set Ω of issues an agent is aware of. Preferences also evolve. This may be because Ω evolves or because the agent is persuaded to change its preferences. Thus the agent's internal theory T, which includes its preferences in ML and a set of other formulae in L modelling the domain, must be explicitly represented in the agent's state. In this model we do not impose any specific requirements on T. Hence the following definition:

Definition 3. A **Negotiation State** for an agent a at time t is any 3-tuple $s = \langle \Omega, T, H \rangle$, where

- Ω is a finite collection of negotiable issues.
- $T \subseteq L \cup ML$, is a theory in the common languages.
- H, the negotiation history, is the set of all negotiation threads involving agent a. That is, $H = \{\vartheta_{i \leftrightarrow a} | i \in Agents\}$.

All possible negotiation states for agent a will be denoted by S_a. As an illustration of how these notions are used, consider the following example:

Example 1. The CSD agent is negotiating with a VC_i agent for the Vet_Customer service for company A. The CSD agent proposes that the service be completed for $\pounds 10$ and should take 24 hours. VC_i responds that company A is known to be in financial difficulty and therefore a more time consuming and expensive vetting should be undertaken (Figure 2, id 8). Moreover, in order to meet the deadline, VC_i will need to delay the vetting of another BT customer (company B) for which an agreement has already been reached. This dialogue may be represented in CL as the sequence:

1. `offer`$(CSD, VC_i, Company = A \wedge price = \pounds 10 \wedge time = 24h, t_1)$
2. `appeal`$(VC_i, CSD, Company = A \wedge price = \pounds 20 \wedge time = 48h,$
   $Financial\_Status = bad \wedge Quality\_vetting = high, t_2)$
3. `appeal`$(VC_i, CSD, Company = B \wedge delay = 24h,$
   `accept`$(VC_i, CSD, Company = A \wedge price = \pounds 20 \wedge time = 48h, t_2), t_3)$

This example shows how the range of issues Ω involved in the negotiation is extended (the delaying of the vet customer service for company B) and how new information (the fact that company A is known to be in financial difficulty) can be brought to bear. This revelation of information means that the CSD agent extends its domain theory T (to include the fact that A may not be creditworthy). ■

3.3 Persuasive agents

As the previous example already showed, the illocutionary acts in CL built from I_{pers} allow arguments to be made in support of a deal. The basic building block for argumentation is $\texttt{appeal}(a, b, \xi, [not]\varphi, t)$ where $a, b \in Agents$, $\xi \in L \cup ML$, and $\varphi \in L \cup ML \cup CL$. This is read as "agent a wants agent b to add ξ to its current theory with argument $[not]\varphi$ supporting it". The other persuasive illocutionary acts, $\texttt{threaten}(a, b, [not]\psi_1, [not]\psi_2, t)$ and $\texttt{reward}(a, b, [not]\psi_1, [not]\psi_2, t)$ with $\psi_1, \psi_2 \in CL$, can contain arguments as long as ψ_1 and/or ψ_2 are appeals, or, recursively, contain appeals.

The interpretation of a persuasive argument for a formula determines whether the hearing agent changes its theory. To make a choice the agent considers the (possibly conflicting) arguments coming from other agents, and from itself, as proofs generated by its own theory. In our domain, and in other work on MAS [2], the social role between the agents is a determining factor in deciding which argument should be preferred. Hence, an authority relation is derived from the social roles and this is then used as the mechanism for comparing arguments. Precisely which social roles correspond to a power relation between the agents depends on the particular domain. In this scenario, for example, the role 'contractor' determines a power relation between the CSD agent and the vetting companies. To build a directed graph representing the authority that one agent has over another, we take the labelled graph associated with the social relation R, remove the links labelled with non-power roles, and add the necessary links to make the relation transitive. Hence the following definition:

Definition 4. Given a Dialogical Framework $DF = \langle Agents, Roles, R, L, ML, CL, Time\rangle$ and a set of authority roles $Power \subseteq Roles$, we define the *authority graph*, $AG \subseteq Agents \times Agents$, for DF as:

1. If $R(a, b) \in Power$ then $(a, b) \in AG$
2. If $(a, b), (b, c) \in AG$ then $(a, c) \in AG$

We say an authority graph is well defined if it is acyclic.

The authority graph encodes the authority relation—or lack of it, since in general AG is not totally connected—between any two agents. Now, our position is that in this domain the 'power' of an argument is determined solely by the authority of the agents which contribute formulae to its construction. Hence, it is necessary to extend the notion of authority from a relation between agents, as captured in the authority graph, to a relation over sets of agents which will be used to establish which arguments to prefer. There are two obvious ways of defining such a relation. We say that a set of agents A has *lower minimum authority* than B, $A \sqsubset_{\min} B$, if and only if for all $b \in B$ there exists $a \in A$ such that $(b,a) \in AG$. And that A has *lower maximum authority* than B, $A \sqsubset_{\max} B$, if and only if for all $a \in A$ there exists $b \in B$ such that $(b,a) \in AG$. Thus, intuitively, the order $\sqsubset_{\min}$ assumes that if any formula used in the argument was proposed by somebody low in the authority graph the argument is weak, while $\sqsubset_{\max}$ assumes that as soon as any formula in the argument is proposed by somebody high in the authority graph the argument is strong. Obviously other authority relations might also be proposed. From now on we refer to any authority relation by the symbol $\sqsubset$.

In its most general form an argument is a proof for a formula [1]. We assume that all agents share the same deductive systems for L ($\vdash_L$) and ML ($\vdash_{ML}$). Hence, in this restricted context, a proof can be represented as the conjunction of all the formulae used in it because it can be reconstructed by the agent receiving it. An argument is then a formula $\varphi \in L \cup ML \cup CL$ that might be constructed from atomic formulae present initially in the theory of the agent or obtained in previous negotiation encounters from different agents. Assuming the existence of a function $Support : L \cup ML \cup CL \to 2^{Agents}$ that gives the agents whose formulae are used in the construction of an argument, or the agent that uttered the illocution when $\varphi \in CL$. We can use the social role of those agents to decide how forceful an argument is.

Fundamental to this view of decision making is the idea that one argument may attack another [3]. We represent the fact that an argument Arg supports a formula φ as a pair (Arg, φ) and the fact that the argument pair (Arg_1, φ_1) attacks (Arg_2, φ_2) by $Attacks((Arg_1, \varphi_1), (Arg_2, \varphi_2))$. The precise meaning of $Attacks$ depends strongly on the concrete languages L and ML being used. For the purpose of this paper we follow Dung [3] in assuming that it is a primitive notion, because our focus is on how to resolve the effect of an attack no matter how it is defined.

Definition 5. Given the two argument pairs (Arg_1, φ_1) and (Arg_2, φ_2) such that $Attacks((Arg_1, \varphi_1), (Arg_2, \varphi_2))$ then (Arg_1, φ_1) will be preferred to (Arg_2, φ_2), which we write as $(Arg_2, \varphi_2) \prec (Arg_1, \varphi_1)$, if and only if $Support(Arg_2) \sqsubset Support(Arg_1)$. When $(Arg_2, \varphi_2) \not\prec (Arg_1, \varphi_1)$ and $(Arg_1, \varphi_1) \not\prec (Arg_2, \varphi_2)$ we say that an agent is indifferent with respect to the arguments—and denote this by $(Arg_1, \varphi_1) \sim (Arg_2, \varphi_2)$.

The agents use argumentation as the means to decide how to interpret incoming and generate outgoing illocutions. On receiving an argument pair (Arg_1, φ_1) that is not attacked by any argument pair (Arg_2, φ_2) built from its current theory, an open-minded agent may simply add the argument Arg_1 and the formula φ_1 to its theory. In contrast, a more conservative agent may not accept a proposition unless it comes from a higher authority. When $Attacks((Arg_1, \varphi_1), (Arg_2, \varphi_2))$ the most preferred (in the sense defined above) argument pair is kept. If $(Arg_1, \varphi_1) \sim (Arg_2, \varphi_2)$ some additional criteria must be applied to decide which to keep, for instance epistemic entrenchment [4].

Example 2. The DD and SD agents are negotiating over the Survey_Customer_Site service. DD proposes that the service should be completed within 24 hours. SD indicates that one of its surveyors was planning to go on holiday and so the survey will take 48 hours (Figure 2, id 9). DD indicates that it must have the service completed within 24 hours. In CL this is expressed as:

1. `offer`$(DD, SD, time = 24h \wedge service = Survey\_Customer\_Site, t_1)$

2. $\texttt{appeal}(SD, DD, time = 48h, surveyor(Smith) \wedge holiday(Smith), t_2)$
3. $\texttt{appeal}(DD, SD, time = 24h, time = 24h, t_3)$

In this example, SD issues an appeal to DD for more time to complete the survey service. DD rejects this argument saying the service must be completed within 24 hours. SD now has two arguments that attack one another: $Attacks((surveyor(Smith) \wedge holiday(Smith), time = 48h), (time = 24h, time = 24h))$. It resolves them by referring to its authority graph which indicates that the authority of DD's argument is more powerful than its own (since DD is its boss, that is, $(DD, SD) \in AG$) and therefore it must do whatever is necessary to ensure the service is completed within 24 hours. That is, $Support(surveyor(Smith) \wedge holiday(Smith)) = \{SD\}$, $Support(time = 24h) = \{DD\}$ and given that $(DD, SD) \in AG$ we have that $(surveyor(Smith) \wedge holiday(Smith), time = 48h) \prec (time = 24h, time = 24h)$ because in our example $\{SD\} \sqsubset \{DD\}$ (using either of the measures mentioned above). ■

3.4 Interpretation and Generation of Illocutions

For pragmatic reasons, we separate the definition of the semantics of illocutions into two different operations, I and G (see examples 3 and 4). The former implements the negotiation-state transition associated with hearing a given illocution, while the latter determines the illocutionary action to be taken in a particular state.

The underlying idea is that any illocution may introduce new issues into a negotiation, while appeals may, in addition, modify the preference relationships and the agent's theory. However, the actual effect of an illocution depends on the agent's interpretation of the utterances it receives. This interpretation process is highly domain-specific and is also dependent upon the internal structures present in the agent architecture. For this reason, we illustrate how our framework can be used to define a comparatively simple open-minded agent. Naturally this does not prescribe how all agents should behave, but rather exemplifies the concepts of our model which can be used to define many other types of agent.

The illocution interpretation function I for an open-minded agent is based on the following intuitions:

- Every illocution extends the corresponding thread in the negotiation history[6]. In this way, for example, complete illocutionary histories allow agents with total recall to be modelled. Forgetful agents can then be modelled by discarding part of the negotiation thread.
- All illocutions may introduce new issues into the negotiation.
- Appeals may change an agent's preference relationship. They may change the theory as well by extending it with the formulae of the argument in the appeal, provided that the current theory cannot build attacking arguments for the appeal.

[6] However, we do not update agents' theories in this minimal semantics because we wish to keep the interpretation of illocutions reasonably neutral with respect to the agents' internal architectures.

Example 3. Open-minded Interpretation. Given a communication language CL, a dialogical framework DF, and the set of all possible negotiation states S_b for an agent b, the interpretation function for an *open-minded agent* is defined by $I : CL \times S_b \times DF \rightarrow S_b$ such that—having $s = (\Omega, T, H)$, $H = \{\vartheta_{i \leftrightarrow b} | i \in Agents\}$, and '^' representing concatenation— we have[7]:

1. $I(\iota(a, b, \delta, t), s, df) = (\Omega \cup issues(\delta), T, H - \vartheta_{b \leftrightarrow a} + \vartheta'_{b \leftrightarrow a})$
   with $\iota \in I_{nego}$; $\vartheta'_{b \leftrightarrow a} = \vartheta_{b \leftrightarrow a} \hat{} \iota(a, b, \delta, t)$
2. $I(\texttt{threaten}(a, b, [not]\psi_1, [not]\psi_2, t), s, df) =$
   $(\Omega \cup issues(\psi_1) \cup issues(\psi_2), T, H - \vartheta_{b \leftrightarrow a} + \vartheta'_{b \leftrightarrow a})$
   with $\vartheta'_{b \leftrightarrow a} = \vartheta_{b \leftrightarrow a} \hat{} \texttt{threaten}(a, b, [not]\psi_1, [not]\psi_2, t)$
3. $I(\texttt{reward}(a, b, [not]\psi_1, [not]\psi_2, t), s, df) =$
   $(\Omega \cup issues(\psi_1) \cup issues(\psi_2), T, H - \vartheta_{b \leftrightarrow a} + \vartheta'_{b \leftrightarrow a})$
   with $\vartheta'_{b \leftrightarrow a} = \vartheta_{b \leftrightarrow a} \hat{} \texttt{reward}(a, b, [not]\psi_1, [not]\psi_2, t)$
4. $I(\texttt{appeal}(a, b, \xi, [not]\varphi, t), s, df) = (\Omega', T', H - \vartheta_{b \leftrightarrow a} + \vartheta'_{b \leftrightarrow a})$
   with $\vartheta'_{b \leftrightarrow a} = \vartheta_{b \leftrightarrow a} \hat{} \texttt{appeal}(a, b, \xi, [not]\varphi, t)$;
   if no (Arg, ψ) built from T such that $Attacks(([not]\varphi, \xi), (Arg, \psi))$
   then $\Omega' = \Omega \cup issues(\xi) \cup issues(\varphi)$;
   if $\varphi \in L \cup ML$ **then** $T' = T + \xi + \varphi$ **else** $T' = T + \xi$
   else $\Omega' = \Omega; T' = T$

■

Finally, an agent a's specification must include a way of computing the next illocution to be uttered in the negotiation thread. That is a function $G : S_a \times DF \rightarrow CL$ needs to be defined. This function must conform with the protocol depicted in Figure 3 and can conveniently be represented as a collection of condition-action rules, where the action is an illocutionary action. How an agent chooses which illocution to utter depends on many factors: the history of the negotiation, the active goals of the agent, or its theory, and it also depends on the way that particular agent interprets those illocutions. The following example illustrates a simple negotiation dialogue between two agents and contains a fragment of a G function.

Example 4. We use an expanded version of the argument presented in Example 2 to illustrate specific instances of illocution generation and interpretation functions. Given the two initial illocution interchanges:

1. $\texttt{offer}(DD, SD, time = 24h \wedge service = Survey\_Customer\_Site, t_1)$
2. $\texttt{appeal}(SD, DD, time = 48h, surveyor(Smith) \wedge holiday(Smith), t_2)$

We show two decisions taken by two different types of agent; an 'authoritarian' DD agent which exploits its social power (and threatens to inform the company chairman that SD did not agree to complete the task within 24h), and a 'conciliatory' DD agent which resorts to an explanatory appeal (that it is company policy that quotes must be handled within 24h):

3.1 **Authoritarian**: $\texttt{threaten}(DD, SD, not\ \texttt{accept}(SD, DD, time = 24h, t_3),$
$\texttt{appeal}(DD, Chairman, not\ \texttt{accept}(SD, DD, time = 24h, t_3), t_4))$

[7] An alternative way of looking at the interpretation of illocutions is as programs that transform one state into another. A natural formalism for that interpretation is Dynamic Logic [12].

3.2 **Conciliatory**: $\texttt{appeal}(DD, SD, time = 24h, BT\_Policy\_Time = 24h, t_3)$

The G function of an 'obedient' SD agent that, whenever possible, does what it is told could include the following decision rules where 'self' represents the agent interpreting the illocution:

if $\check{\vartheta}_{x \leftrightarrow self} = \texttt{threaten}(x, self, not\ \texttt{accept}(self, x, \delta), \psi_2)$ **and** $(x, self) \in AG$
 and $can\_do(\delta)$ **then** $\texttt{accept}(self, x, \delta)$
if $\check{\vartheta}_{x \leftrightarrow self} = \texttt{threaten}(x, self, not\ \texttt{accept}(self, x, \delta), \psi_2)$ **and** $(x, self) \in AG$
 and not $can\_do(\delta)$ **then** $\delta' = compute\_counter\_offer(s, DF)$;$\texttt{offer}(self, x, \delta')$
if $\check{\vartheta}_{x \leftrightarrow self} = \texttt{appeal}(x, self, \xi, \varphi)$ **and** $\psi \rightarrow \neg\varphi \in T$ **then** $\texttt{appeal}(self, x, \neg\varphi, \psi)$

Assuming that $can\_do(time = 24h \wedge service = Survey\_Customer\_Site)$ is true, by subcontracting the task say, the dialogue with the authoritarian DD ends with:

4.1 $\texttt{accept}(SD, DD, time = 24h \wedge service = Survey\_Customer\_Site, t_4)$

On the other hand, if we assume that the rule $BT\_Policy\_Time = 24h \leftrightarrow Fully\_staffed$ is true and DD utters 3.2, the agent could reply with:

4.2 $\texttt{appeal}(SD, DD, not\ (BT\_Policy\_Time = 24h), not\ Fully\_staffed)$ ■

To further illustrate the power of our framework, Figure 4 shows the representation in CL of the arguments presented in Figure 2.

Id	Dialogue
1	$\texttt{appeal}(CSD, VC_i, \texttt{offer}(VC_j, CSD, \delta), true)$, $\texttt{threaten}(CSD, VC_i, not\ \texttt{offer}(VC_i, CSD, \delta), \texttt{withdraw}(CSD, VC_i))$
2	$\texttt{threaten}(CSD, VC_i, not\ \texttt{offer}(VC_i, CSD, \ldots \wedge time < limit)$, $not\ \texttt{request}(CSD, VC_i, Future^a))$ ^{a} $Future$ is an universally quantified variable over the future instants in $Time$.
3	$\texttt{threaten}(DD, SD, not\ \texttt{acccept}(SD, DD, \ldots \wedge time < limit)$, $\texttt{appeal}(DD, Boss_{SD}, \psi^a, not\ \texttt{acccept}(SD, DD, \ldots \wedge time < limit)))$ ^{a} ψ expressing the fact that the deadline has been missed.
4	$\texttt{reward}(CSD, DD, \texttt{accept}(DD, CSD, \delta), \texttt{appeal}(CSD, OurBoss, \psi, \texttt{accept}(DD, CSD, \delta)))^a$ ^{a} $\delta = \ldots Vet = Customer_i \wedge time < limit$. The reward consists of passing the information to our boss. ψ represents the satisfaction of $Customer_i$.
5	$\texttt{reward}(CSD, VC_i, \texttt{accept}(VC_i, CSD, \ldots \wedge time = k \wedge \ldots), \texttt{request}(CSD, VC_i, \Delta, Future))^a$ ^{a} Δ stands for a deal, and $Future$ stands for an instant in the future.
6	$\texttt{appeal}(CSD, VC_i, time = t \wedge cost = c, \texttt{accept}(VC_i, CSD, \ldots \wedge time = t \wedge cost = c, Before^a))$ ^{a} *Before* represents a previous instant in $Time$.
7	$\texttt{appeal}(CSD, DD, time = 48h, BT\_policy\_time = 48h)$
8	$\texttt{appeal}(VC_i, CSD, time = high, Financial\_status = trouble \wedge Quality\_vetting = high)$
9	$\texttt{appeal}(DD, CSD, time > t_{normal}, surveyor(Smith) \wedge holiday(Smith))$
10	$\texttt{appeal}(SD, DD, time > t_{normal}, Number\_premises = High)$

Fig. 4. Formalisation of the arguments presented in Figure 2.

4 Related Work

Much of the existing work on agent-based negotiation is rooted in game theory, e.g. [15]. Although this approach has produced significant results, and has been successful in many negotiation domains, it embodies a number of limiting assumptions about the agents' knowledge and utility functions. Even when this approach is extended, as in [11], to cope with conditions that change over time, it does not address the problem of how these changes can be accomplished by one agent influencing another, nor does it cope with the problem of introducing new issues into negotiations. Changing preferences through persuasion, in multi-agent systems, was addressed in Sycara's seminal work on labour negotiation [18], work extended and formalised by Kraus *et al.* [10]. However, this work is set within the context of a particular agent architecture, assumes a fixed and shared domain theory, and deals with five particular types of argument (threats, rewards, appeals to precedent, appeals to prevailing practice, and appeals to self-interest). Furthermore, Kraus *et al.* do not deal with the introduction of new issues or imperfect rationality. In contrast, our model accommodates partial knowledge, imperfect rationality and the introduction of new negotiation issues—which are relevant features in many application domains—while only imposing minimal requirements on agents' internal states and using a general rhetorical language.

We should also acknowledge the differences between our work and the use of argumentation to explain how a single agent reasons. In the former, an agent argues with itself to establish its beliefs. In our work arguments are used by one agent in order to change another agents' beliefs and actions. The other important difference is that the mechanism for resolving conflicts between arguments in single agent argumentation is often built into the logical language in which arguments are constructed and is based upon some intuitive notion of what is correct in the world at large. In contrast, we keep this mechanism at the meta-level and ground it in knowledge about the domain. This has the dual advantage of ensuring that conflicts are resolved in a way that is known to be suitable for our domain whilst allowing new conflict resolution mechanisms to be easily fitted into the model in different domains.

5 Conclusion

This paper has introduced a novel framework for describing persuasive negotiations between autonomous agents. This provides a sound foundation for building specific artificial agents by instantiating the generic components such as L, ML and T. The framework has been strongly influenced by our experience of business process management applications and this makes us confident that it can capture the needs of other real world applications. However, we realise that there are a number of issues which require further investigation. Firstly there is the matter of how expressive CL is required to be. For instance, at the moment an agent can only make threats and promises about illocutionary actions (e.g. to tell somebody about something). It is also desirable for non-illocutionary actions to be the consequence of a threat or promise. Similarly, while appeals could be used to model a wide range of illocutions, it may be useful to characterise subtly different types of illocution through more refined interpretation and generation functions. Secondly, we have reflected an agent's preferences, and the changes in those preferences, simply as sentences and updates in the agent's theory T. Further work is required to tie these preferences to notions of rationality, in particular to standard ideas of expected utility. Finally, we make the simplifying assumption that negotiating agents have a common notion of deduction. This may be inadequate for some domains, in which case it will be necessary for agents to be able to discuss what rules of inference are appropriate.

References

1. S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proc 9th Conf on Uncertainty in AI*, pages 411–419, 1993.
2. C. Castelfranchi. Social Power: A Point missed in Multi-Agent, DAI and HCI. In Y. Demazeau and J. P. Müller, editors, *Decentralised AI*, pages 49–62. Elsevier, 1990.
3. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
4. P. Gärdenfors. *Knowledge in Flux*. MIT Press, Cambridge, MA, 1987.
5. F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics (or: How we can do without modal logics). *Artificial Intelligence*, 65:29–70, 1994.
6. W. D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.
7. T. R. Gruber. The role of common ontology in achieving sharable, reusable knowledge bases. In J. A. Allen, R. Fikes, and E. Sandewall, editors, *Proc. of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning*, 1991.
8. N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *International Journal of Cooperative Information Systems*, 5(2&3):105–130, 1996.
9. M. Karlins and H. I. Abelson. *Persuasion*. Crosby Lockwood & Son, London, UK, 1970.
10. S. Kraus, M. Nirkhe, and K. Sycara. Reaching agreements through argumentation: a logical model (preliminary report). In *DAI Workshop'93*, pages 233–247, 1993.
11. S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75:297–345, 1995.
12. P. Noriega and C. Sierra. Towards layered dialogical agents. In *Proceedings of the ECAI'96 Workshop Agents Theories, Architectures and Languages, ATAL'96*, number 1193 in LNAI, pages 157–171, 1996.
13. S. Parsons and N. R. Jennings. Negotiation through argumentation—a preliminary report. In *Proc. Second Int. Conf. on Multi-Agent Systems*, pages 267–274, 1996.
14. A. S. Rao and M. P. Georgeff. BDI agents: From Theory to Practice. In *Proc 1st Int Conf on Multi-Agent Systems*, pages 312–319, 1995.
15. J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, Cambridge, USA, 1994.
16. C. Sierra, P. Faratin, and N. R. Jennings. A service-oriented negotiation model between autonomous agents. In *MAAMAW'97*, number 1237 in LNAI, pages 17–35, 1997.
17. R. G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Trans on Systems, Man and Cybernetics*, 11(1):61–70, 1981.
18. K. P. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.
19. D. N. Walton. *Informal Logic*. Cambridge University Press, Cambridge, UK, 1989.