Arguing for Gaining Access to Information

Sylvie Doutre
IRIT–Univ. Toulouse 1
2 rue du doyen
Gabriel Marty
F–31042, Toulouse
France
sylvie.doutre@irit.fr

Peter McBurney
Dpt. of Computer Sc.
Univ. of Liverpool
Liverpool L69 3BX
United Kingdom
p.j.mcburney@csc.liv.ac.uk

Laurent Perrussel and
Jean-Marc Thévenin
IRIT–Univ. Toulouse 1
2 rue du doyen
Gabriel Marty
F–31042, Toulouse
France
{perussel,thevenin}@univ-tlse1.fr

ABSTRACT

This paper presents a formal protocol for agents engaged in argumentation over access to information sources. Obtaining relevant information is essential for agents engaged in autonomous, goal-directed behavior, but access to such information is usually controlled by other autonomous agents having their own goals. Because these various goals may be in conflict with one another, rational interactions between the two agents may take the form of a dialog, in which requests for information are successively issued, considered, justified and criticized. Even when the agents involved in such discussions agree on all the arguments for and the arguments against granting access to some information source, they may still disagree on their preferences between these arguments.

To represent such situations, we design a protocol for dialogs between two autonomous agents for seeking and granting authorization to access some information source. This protocol is based on an argumentation dialog where agents handle specific preferences and acceptability over arguments. We show how this argumentation framework provides a semantics to the protocol dedicated to the exchange of arguments, and we illustrate the proposed framework with an example in medicine.

1. INTRODUCTION

The problem of gaining information from multiple sources is one of the key problems for the definition of autonomous agent. In this paper, we advocate that giving to agents the ability to negotiate access to information brings more flexibility to multi-agent systems for handling information-seeking problem. Introducing negotiation access authorization through a persuasion dialog allows to enhance the trust aspect in multi-agent systems [16].

We show how two agents, a client and a server, may dialog so that the client tries to get access to information held by the server while the server tries to convince the client that it cannot give it the access. In that context, gaining access to information can be viewed as an argumentation dialog [14, 13, 12] where agents exchange arguments and counter-arguments in order to set common agreements about authorizations. Agents present arguments which represent their own point of view, i.e. arguments they consider as the more persuasive. Multi-agent dialog based on argumentation [12, 11, 15, 10] for information-seeking [4, 17] as well as preference-based argumentation systems [3, 2, 1, 5] have already been studied. These preferences over arguments help agents to characterize their own acceptable arguments which represent the foundation on which agents accept or not to change authorizations: that is, agents controlling access to information consider to be convinced as long as their acceptable arguments against giving permission have not been sufficient to convince their opponent.

There are very few papers dealing with the problem of how agents may control the access [6, 7] in the context of an argument-based negotiation framework. None of them describe this process in the context of an explicit link between permissions and arguments for and against these permissions. This explicit link enables agents to justify why they provide or do not provide information.

In this paper, we propose a protocol of dialog for getting access to information: that is, a client argues for getting the permission to access information while the server argues against the argument proposed by the server. The proposed primitives of the agent communication language are based on FIPA-ACL oriented performatives [9] which are widely accepted for describing agents dialogs. This protocol is defined in a formal way. A key issue is that the client and the server select and evaluate the received arguments according to their own notion of acceptability: for instance if the server handles preferences over arguments, it evaluates if the received argument is more convincing than the arguments that backed the refusal of access. The contribution brought by this formalization is twofold: a formal description of the different steps that may occur in the dialog and a semantics of the protocol in terms of multiple preference-based argumentation systems.

The paper is organized as follows. In section 2 we present a motivating example. In section 3 we present the formal framework for representing argumentation-based dialog. In section 4 we describe the protocol that rules the dialog, its characteristics and properties. In section 5 we revisit the initial motivating example and express it in a formal way. We conclude the paper in section 6.

2. A MOTIVATING EXAMPLE

Robert is a British businessman visiting Brussels for a meeting. During his visit he becomes ill and is taken unconscious into hospital. The staff of the hospital suspect Robert has had a heart attack and seek to prescribe appropriate drugs for his condition. Unfortunately the safe choice of drugs depends upon various factors, including prior medical conditions that Robert might have and other

drugs he may be taking. The hospital's agent is given the goal of finding out the required information about Robert, from the agent representing his London doctor.

In order to gain access to information about Robert, the agent of Brussels Hospital establishes the following dialog with the London agent:

Agent of Brussels Hospital: I would like to dialog with the agent of Robert's British doctor; I request Robert's health record.

London agent: I cannot provide you Robert's health record because Robert has only given his British doctor limited consent to pass on his personal information (argument A_1).

Brussels agent: This record could possibly include information that could affect the treatment of Robert's heart failure. I request it, Robert's life may be at stake (argument A_2)!

London agent: I cannot divulge this information, because British law prohibits passing on information without the consent of the provider of the information (argument A_3).

Brussels agent: EC law takes precedence over British law when it would be in the interests of the owner to divulge the information (argument A_4). You should allow me to access the record.

London agent: Only Robert could decide what would be in his interests (argument A_5).

Brussels agent: Robert's doctor owes a duty of care to Robert and, should he die, the doctor might be sued by his family, or the Brussels hospital, or both (argument A_6).

London agent: OK. I provide you the requested record: Robert's history of diabetes is...

As we can see, there are numerous key issues in this dialog. First the Brussels and London agents set an agreement about information that is considered: setting/getting access to some information. Second, London agent interacts with Brussels agent because it *controls* information about Robert's health record. Next London agent presents an argument A_1 which attacks the Brussels agent's request: argument A_1 is an argument against giving permission to Brussels agent; in other words *permissions are argued*. Then Brussels and London agents exchange counter-arguments (A_2...). It follows that they both *share* the same set of arguments (they understand each other) and they also share the notion of attack. Indeed they agree in an implicit way that the proposed argument by the opponent attacks the previously proposed argument. At the end of the dialog, London agent accepts the final Brussels' argument A_6. It follows that in an implicit way, London agent agrees that argument A_6 is an acceptable argument which supports the permission in favor of Brussels. Consequently, London agent *changes the permission* and *provides* the requested information to Brussels agent.

In the following we describe a formal system that embed this kind of dialog.

3. FORMAL FRAMEWORK

In this section we describe in a formal way the main concepts that have been previously introduced: access rights, primitives of dialogs and arguments which help us to specify the negotiation process.

First we give some preliminaries. Let Ag be the set of agent identifiers (id). In the following an agent id is represented by a lower case Roman letter (x, y, ...). We assume the information requested is identified by lower case Greek letters (ϕ, ψ...). Let Inf be the set of all possible information ids. This information may be any of: a data record (e.g., one patient's record); a database (e.g., records of many patients); or even the protocol for another dialog (e.g., a client may first request a server to enter into a second dialog, which requires authorization to engage in). The actual content corresponding to information ϕ is denoted by $\langle \mathrm{content}\ \phi \rangle$.

3.1 Access to information

This paper focuses on how an argumentation process can be used for getting access and, according to our goal, we only consider binary permission (access or not) and we do not consider fine-grain authorizations such as read and write accesses. The permission a participant x has to access the content of information ϕ is denoted by a function $\mathrm{perm}(x, y, \phi)$: $\mathrm{perm}(x, y, \phi) = 1$ (respectively 0) stands for agent x can give (respectively cannot give) to agent y the content of information ϕ. Formally,

$$\mathrm{perm} : \mathsf{Ag} \times \mathsf{Ag} \times \mathsf{Inf} \mapsto \{0, 1\}$$

Permission is closely linked to the notion of control. An agent can define permissions about information ϕ only if it actually controls the access to ϕ. In the following we represent this notion of control through a function $\mathrm{control}$ which associates agents and pieces of information:

$$\mathrm{control} : \mathsf{Ag} \mapsto 2^{\mathsf{Inf}}$$

By splitting control and permission we avoid the problem that an agent gives itself permissions to all pieces of information.

Example 1 *Let us consider the initial intuitive example. Let* $\mathsf{Ag} = \{b, l\}$ *s.t.* b *is Brussels agent id and* l *is London agent id; let* ρ *stands for "Robert's health record" and thus* $\mathsf{Inf} = \{\rho\}$*. London's control and permission are defined as follows:*

$$\mathrm{control}(l) = \{\rho\} \qquad \mathrm{perm}(l, b, \rho) = 0$$

3.2 Primitives of dialogs

This is the syntax of a dialog system for information-seeking which requires permission to access the information.

Participants There are two participants, a *Client* (requesting information), and a *Server* (controlling access to some information, which it may or may not agree to provide).

Dialog goal The Client has the following goal prior to the start of the interaction: to obtain from the Server all the information it needs, using persuasion if necessary. The Server has the following goal prior to the start of the interaction: To provide information to the Client according to the level of access permission the Client has.

Context Client and Server may have disjoint knowledge bases. The knowledge base of the Server includes information about the access permissions which each Client has, which may differ by the information concerned.

Arguments We assume the arguments exchanged by agents are represented by upper Roman letters (A, B,...). The internal structure of an argument is left abstract.

Communication language The primitives of the dialogs presented hereafter are mainly based on [9]. The minimum locutions needed for a dialog between Client x and Server y are:

$\mathbf{OpenDialogue}(x, y)$ Client x indicates to Server y that it wants to enter into a dialog.

$\mathbf{Ask}(x, y, \phi)$ Client x asks Server y to provide it with some information ϕ.

$\mathbf{Tell}(y, x, \langle \text{content } \phi \rangle)$ Server y provides Client x with the actual content of information ϕ.

$\mathbf{DontTell}(y, x, \phi)$ Server y indicates to Client y that it cannot provide x with information ϕ.

$\mathbf{EndDialogue}(x, y)$ Agent x indicates to Agent y that it wants to leave the dialog.

In case Client x would not have the permission to access information ϕ, an argumentation dialog about the addition of this permission in Server y's knowledge base is engaged. In this case, a locution for arguing about the permission related to requested information ϕ may be uttered:

$\mathbf{Argue}(z, t, \iota, A)$ Agent z gives to agent t an argument A stating why the permission should be equal to value ι. In the following, $\mathbf{Argue}(x, y, 1, A)$ stands for Client x gives an argument A to Server y as to why it should have the permission (to access ϕ) while $\mathbf{Argue}(y, x, 0, A)$ stands for Server y gives to Client x an argument A as to why x cannot have access (to information ϕ).

Let loc be a locution standing between a client and a server. In the following $agents(loc)$ stands for the set composed of the agents ids that appear in the locution.

3.3 Argumentation framework

In our proposal we require an argumentation framework that enables agents to share the same set of arguments and the same defeat relation between arguments. In addition, each agent should be able to determine its own set of acceptable arguments. Arguments and defeat relation can be represented using the system proposed by [8]. Handling preferences over arguments is one of the simplest way for representing different points of view over the same set of arguments. [1] has presented an extension of [8] that takes into account a unique preference relation. [5] has presented another extension where values are associated to arguments and each agent defines its own set of preferences over these values, and thus over arguments. Work has also already been done to characterize acceptable arguments in the context of multiple preference relations [3, 2]: these works show how to integrate the different relations into a new and unique resulting relation. In our context, we do not need to integrate all specific preference relations into a single one nor we need to specify how preferences are defined: our aim is rather to define a framework where an agent facing an argument can propose a counter-argument based on its preferences which it deems acceptable. For instance, let us consider our medicine example: argument A_2, "This record could possibly include information that could affect the treatment of Robert...", is acceptable for the Brussels agent, but not for the London agent. At this stage, we do not need to enforce the usage of a specific notion of acceptability. Hence each agent evaluates the set of arguments with respect to its own notion of acceptability and its own set of preferences: the acceptability relation combined with the preferences represents the first part of its policy for permission negotiation (the second part is represented by the protocol—see section 4). Formally, we obtain the following definition:

Definition 1 (MPAF) *A* Multiple preference-based argumentation framework *(MPAF) is a tuple*

$$\langle \mathsf{Arg}, \Re, \bigcup_{x \in \mathsf{Ag}} \geqslant_x, \text{acceptable} \rangle$$

where:

- Arg *is a set of arguments,*
- $\Re$ *is a defeat relation:* $\Re \subseteq \mathsf{Arg} \times \mathsf{Arg}$,
- $\bigcup_{x \in \mathsf{Ag}} \geqslant_x$ *is a set of preference relations s.t.* $\geqslant_x$ *stands for the preference relation associated to agent* x *and each relation* $\geqslant_x$ *is a partial pre-order.*
- acceptable *is a function which maps agent ids to a subset of* Arg *which characterizes acceptability,* $\text{acceptable} : \mathsf{Ag} \mapsto 2^{\mathsf{Arg}}$. $\text{acceptable}(x)$ *stands for the acceptable set of arguments associated to agent* x. *Each set* $\text{acceptable}(x)$ *is a subset of* Arg *defined w.r.t. the defeat relation* $\Re$ *and preference relation* $\geqslant_x$.

The strict order associated with $\geqslant_x$ is denoted by $>_x$. $A >_x B$ means that agent x strictly prefers argument A to B. The sets of acceptable arguments may be defined by using semantics which characterize the policy of the access control. For instance, in a context where information is sensitive the notion of acceptability will be restrictive, whereas a standard notion of acceptability such as the semantics of [8] or [1] may be considered in a context where information has not a high level of confidentiality.

In this paper, we focus on a usage of an acceptability based on the sets of arguments which are conflict-free [8, 1]. The usage of some specific notion of acceptability does not prevent the general aspect of the framework. We rephrase the notion of *defence* and *admissible* arguments in the context of multiple preferences. An argument A is x-defended by a set of arguments S w.r.t. a preference relation $\geqslant_x$ iff (i) A defends itself (it is preferred to all its counter-argument) or (ii) for every counter-argument B, there exists an argument C which belongs to S such that C defeats B and B is not preferred to C:

Definition 2 (x-defense) *Let* $\mathsf{S} \subseteq \mathsf{Arg}$ *be a set of arguments and* $A \in \mathsf{Arg}$ *be an argument.* A *is* x*-defended by* S *iff* $\forall B \in \mathsf{Arg}$ *s.t.* $(B, A) \in \Re$ *then: (i)* $A >_x B$ *or (ii)* $\exists C \in \mathsf{S}$ *s.t.* $(C, B) \in \Re$ *and* $B \not>_x C$.

The next step is to rephrase the notion of *conflict-free* set of arguments: all the arguments belonging to an x-conflict-free set of arguments are preferred to their counter arguments w.r.t. a preference relation $\geqslant_x$:

Definition 3 (x-conflict-free) *A set* S *is said to be* x*-conflict-free iff* $\forall A, B \in S$, *if* $(B, A) \in \Re$ *then* $A \geqslant_x B$.

The next step is to characterize the admissible arguments.

Definition 4 (x-admissible) *A set* S *of arguments is said to be* x*-admissible iff* S *is* x*-conflict-free and* S x*-defends all its elements.*

The set of acceptable arguments for an agent x is calculated w.r.t. the set of x-admissible arguments. In a classical way we have the skeptical and the credulous methods for characterizing the set of acceptable argument.

Definition 5 (Credulous and Skeptical acceptability) *Let* Arg *be a set of arguments,* $\Re$ *be a defeat relation and* $\geqslant_x$ *be a preference relation. The* credulous *set of acceptable arguments* $\mathsf{Cr}(x)$ *defined w.r.t.* x *is:*

$$\mathsf{Cr}(x) = \{A \in \mathsf{Arg} | \exists \mathsf{S} \text{ s.t. } \mathsf{S} \text{ is } x\text{-admissible and maximal w.r.t. } \subseteq \text{ and } A \in \mathsf{S}\}$$

and the skeptical *set of acceptable arguments* $\mathsf{Sk}(x)$ *defined w.r.t.* x *is equal to:*

$$\mathsf{Sk}(x) = \{A \in \mathsf{Arg} | \forall \mathsf{S} \text{ s.t. } \mathsf{S} \text{ is } x\text{-admissible and maximal w.r.t. } \subseteq, A \in \mathsf{S}\}$$

Example 2 *Let us pursue our review of the intuitive example. The associated MPAF is defined as:*

- *set* Arg *is equal to* $\{A_1, A_2, A_3, A_4, A_5, A_6\}$*;*
- *relation* $\Re$ *is defined as shown on figure 1.*

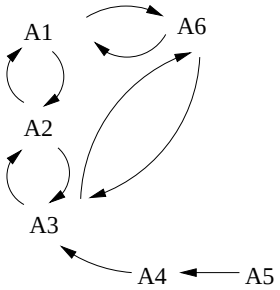

Figure 1: Defeat Relation over arguments set Arg

An arrow between two arguments represents a defeat. That is, argument A_1 *is defeated by arguments* A_2 *and* A_6*; argument* A_2 *is defeated by argument* A_3*. All these defeats are defined in the two directions. Finally argument* A_3 *is defeated by argument* A_4 *and argument* A_5 *defeats* A_4*.*

- *Preference relations for Brussels and London agents are defined as follows (respectively* $\geqslant_b$ *and* $\geqslant_l$*):*

$$\begin{aligned} \geqslant_b &:= A_2 >_b A_1, A_2 >_b A_3, A_6 >_b A_1, A_6 >_b A_3 \\ \geqslant_l &:= A_3 >_l A_2, A_1 >_l A_2 \end{aligned}$$

- *Function* acceptable *is defined as follows. We suppose that both London and Brussels agents use a credulous acceptability. In order to define what arguments are acceptable for each of them, we first calculate the maximal sets which are* b*-admissible and* l*-admissible:*

$$\begin{aligned} \textit{b-admissible set} &= \{A_2, A_5, A_6\} \\ \textit{l-admissible sets} &= \{A_1, A_3, A_5\} \textit{ and } \{A_5, A_6\} \end{aligned}$$

Second, we associate acceptable arguments to each agent

$$\begin{aligned} \text{acceptable}(b) &:= \mathsf{Cr}(b) = \{A_2, A_5, A_6\} \\ \text{acceptable}(l) &:= \mathsf{Cr}(l) = \{A_1, A_3, A_5, A_6\} \end{aligned}$$

3.4 Linking Arguments and Permissions

As shown in the intuitive example, arguments proposed by the Client and the Server are closely connected to their goals. The goal of the Client is to obtain information ϕ while the Server aims at not telling ϕ. In our framework, goals can be rewritten as changing the permission or not. It leads us that we have to connect permissions and arguments; we represent this link by introducing a relation between permissions and arguments which characterizes the notion of *argued permission*.

Definition 6 (argued permission) *An argued permission is a tuple* $\langle A, y, x, \phi, \iota \rangle$ *s.t.* A *is an argument,* y *and* x *are agent ids,* ϕ *is an information and* ι *is the value of the permission (*$\iota \in \{0, 1\}$*).* $\langle A, y, x, \phi, \iota \rangle$ *stands for: Server* y *has the argument* A *in favor (*$\iota = 1$*) or against (*$\iota = 0$*) giving permission to Client* x *to obtain information* ϕ*.*

In fact, it is possible for Server y to consider arguments in favor of giving permission to x about ϕ and at the same time arguments against the same permission. For instance, an agent should not give access to its password for security reason (argument against the permission) and at the same time it may provide it in emergency (argument in favor of the permission). It follows that there is no redundancy to consider a function that describes permissions and arguments in favor or against permissions. However we have to enforce some constraints on permissions by introducing the notion of *consistent* permission. Let us consider Client x, Server y and information ϕ. We claim that a permission defined by y about x and ϕ is *consistent* with a set of argued permissions if (i) y has the control of ϕ (ii) arguments for and against permissions respect the defeat relation and (iii) this permission is "supported" by at least one argument that is acceptable w.r.t. $\text{acceptable}(y)$.

Definition 7 (Consistent permission) *Let* AP *be a set of argued permissions and let* $\mathsf{P} = \{\langle A, y, x, \phi, \iota \rangle\}$ *be the set of argued permission supporting permission* $\text{perm}(y, x, \phi) = \iota$ *and* $\mathsf{C} = \{\langle A, y, x, \phi, 1-\iota \rangle\}$ *be the set of argued permissions against* $\text{perm}(y, x, \phi) = \iota$*. Permission* $\text{perm}(y, x, \phi) = \iota$ *is said to be* consistent *iff:*

1. $\phi \in \text{control}(y)$*;*
2. *argued permissions are constrained by the defeat relation:* $\forall \langle A, y, x, \phi, \iota \rangle \in \mathsf{P}$ *such that for any* y*-admissible set* S *where* $A \in \mathsf{S}$*, if* $\mathsf{C} \neq \emptyset$ *then* $\nexists \langle B, y, x, \phi, 1-\iota \rangle \in \mathsf{C}$*,* $B \in \mathsf{S}$*;*
3. *the following constraint holds between the permission and acceptable arguments:*

$$\text{perm}(y, x, \phi) = \iota \iff \exists \langle A, y, x, \phi, \iota \rangle \in \mathsf{AP} \text{ s.t. } A \in \text{acceptable}(y)$$

The main consequence is that if Server y has adopted a skeptical acceptability relation, then there are no two arguments that belong to the set $\text{acceptable}(y)$ which support opposite permissions.

Proposition 1 *Let* $\text{perm}(y, x, \phi) = \iota$ *be a permission. Let* $\mathsf{C} = \{\langle A, y, x, \phi, 1-\iota \rangle\}$ *be the set of argued permissions against* $\text{perm}(y, x, \phi) = \iota$*. Let* $\mathsf{Sk}(y)$ *be the set of acceptable arguments. No element of* C *belongs to* $\mathsf{Sk}(y)$*:* $\mathsf{C} \cap \mathsf{Sk}(y) = \emptyset$*.*

Notice that the credulous notion of acceptability may entail that acceptable arguments can be in favor and or against a permission at the same time and thus the previous proposition does not hold.

Example 3 *Let us pursue our intuitive example. London agent informs that it cannot provide information about Robert because Robert has only given a limited consent (argument* A_1*). It follows that* A_1 *is an argument against the request of Brussels agent and that tuple* $\langle A_1, l, b, \rho, 0 \rangle$ *is an argued permission. However, this argument is not the only one against giving access to Brussels agent.*

Argument A_3 is also against the authorization while arguments A_2 and A_6 are in favor of the authorization. We get the following set of argued permissions:

$$\mathsf{AP} = \{\langle A_1, l, b, \rho, 0\rangle, \langle A_3, l, b, \rho, 0\rangle, \langle A_2, l, b, \rho, 1\rangle, \langle A_6, l, b, \rho, 1\rangle\}$$

Notice that permission $\mathrm{perm}(l, b, \rho) = 0$ *is consistent: first agent l has control of ρ (see example 1); second, all arguments in favor of permission do not appear at the same time in a same l-admissible set (e.g. $A_1 \in \{A_1, A_3, A_5\}$ but A_2 and A_6 do not belong to $\{A_1, A_3, A_5\}$ and third, there exists an argument involved in an argued permission that is acceptable (e.g. $A_3 \in \mathrm{acceptable}(l)$).*

4. THE PROTOCOL OF NEGOTIATION

In this section, we present a protocol of dialog for information-seeking dialog with permissions. The protocol specifies which locutions may be uttered at different points in a dialog, and so defines the rules governing the use of the locutions previously presented. Now we formally define the concept of dialog. A dialog is a structure that combines access authorizations, a multiple preferences argumentation framework, a set of argued permissions, and a sequence of locution utterances.

Definition 8 (Dialog) *Let $D = \langle \mathrm{control}, \mathrm{perm}, \mathrm{MPAF}, \mathsf{AP}, \sigma\rangle$ be a dialog such that* control *is a function associating agents and information,* perm *is an authorization function,* MPAF *is a multiple preferences argumentation framework,* AP *is a set of argued permissions and σ is a sequence of locutions.*

Let $length$ be a function characterizing the number of elements of a finite sequence of locutions σ and $\sigma[i]$ (s.t. $1 \leqslant i \leqslant length(\sigma)$) represents one element of σ. Now we can express in a formal way the protocol: how the permissions and the arguments are interwoven in order to rule the dialog.

4.1 Structure of the dialog

First we specify that Client x and Server y have to set the dialog. Let $D = \langle \mathrm{control}, \mathrm{perm}, \mathrm{MPAF}, \mathsf{AP}, \sigma\rangle$ be a dialog s.t. σ is a finite sequence: $\exists n$ s.t. $n = length(\sigma)$. In all the following formulas, logical connectors are used w.r.t. their usual meaning.

Formula (S1) states that the dialog should start with an "open" locution.

$$\sigma[1] = \mathbf{OpenDialogue}(x, y) \quad \text{(S1)}$$

Next, formula (S2) specifies that the Client and the Server involved in the dialog should not changed in the whole dialog:

$$\sigma[1] = \mathbf{OpenDialogue}(x, y) \Longrightarrow \forall i(1 < i \leqslant length(\sigma) \Longrightarrow agents(\sigma[i]) = \{x, y\}) \quad \text{(S2)}$$

Formula (S3) states that the dialog should end with an **EndDialogue** locution

$$\sigma[length(\sigma)] = \mathbf{EndDialogue}(x, y) \vee \sigma[length(\sigma)] = \mathbf{EndDialogue}(y, x) \quad \text{(S3)}$$

Finally formula (S4) states that an **OpenDialogue** locution can only occur at the beginning of the dialog and that the dialog can only be closed at the end:

$$\forall i(1 < i < length(\sigma) \Longrightarrow (\sigma[i] \neq \mathbf{OpenDialogue}(x, y) \wedge \sigma[i] \neq \mathbf{EndDialogue}(y, x) \wedge \sigma[i] \neq \mathbf{EndDialogue}(x, y))) \quad \text{(S4)}$$

4.2 Requesting information

After opening the dialog, the Client requests some information ϕ (formula (R1)):

$$\sigma[1] = \mathbf{OpenDialogue}(x, y) \Longrightarrow \sigma[2] = \mathbf{Ask}(x, y, \phi) \quad \text{(R1)}$$

Formula (R2) states that the Server should provide ϕ if the Server control ϕ and the Client has the authorization to access information ϕ:

$$\sigma[2] = \mathbf{Ask}(x, y, \phi) \wedge \phi \in \mathrm{control}(y) \wedge \mathrm{perm}(y, x, \phi) = 1 \Longrightarrow \sigma[3] = \mathbf{Tell}(y, x, \langle \mathrm{content}\ \phi\rangle) \quad \text{(R2)}$$

Formula (R3) states that if the Server has no control over ϕ then it should close the dialog:

$$\sigma[2] = \mathbf{Ask}(x, y, \phi) \wedge \phi \notin \mathrm{control}(y) \Longrightarrow \sigma[3] = \mathbf{EndDialogue}(y, x) \quad \text{(R3)}$$

Formula (R4) specifies that if the Server has actually provided information ϕ then the dialog should be closed by the Client or the Server:

$$\sigma[i] = \mathbf{Tell}(y, x, \langle \mathrm{content}\ \phi\rangle) \Longrightarrow \sigma[i{+}1] = \mathbf{EndDialogue}(x, y) \vee \sigma[i{+}1] = \mathbf{EndDialogue}(y, x) \quad \text{(R4)}$$

Now let us focus on the case which will lead us to the argumentation part of the dialog; that is, where the Server cannot provide information ϕ to the Client. Formula (R5) formally specifies the condition where a **DontTell** locution can be uttered.

$$\sigma[2] = \mathbf{Ask}(x, y, \phi) \wedge \phi \in \mathrm{control}(y) \wedge \mathrm{perm}(y, x, \phi) = 0 \Longleftrightarrow \sigma[3] = \mathbf{DontTell}(y, x, \phi) \quad \text{(R5)}$$

Indeed permissions help the Server to define its answer to the Client. As we will see in the next section, permissions will no longer be used; instead the Server will try to convince the Client to "forget" its request by using argued permissions and arguments and the Client will try in turn to convince the Server to give it permission.

4.3 Arguing for getting permission

In this section we describe the rules that characterize the negotiation stage. Formula (G1) states that argumentation occurs only if the Server does not want to provide information to the Client:

$$\sigma[i] = \mathbf{Argue}(y, x, 0, A) \vee \sigma[i] = \mathbf{Argue}(x, y, 1, A) \Longrightarrow \sigma[3] = \mathbf{DontTell}(y, x, \phi) \quad \text{(G1)}$$

If the Server refuses to answer the Client the argumentation stage is initiated. In this paper, for the sake of conciseness we assume that this stage is initiated by the Server. Formula (G2) specifies that the Server has to motivate its refusal.

$$\sigma[3] = \mathbf{DontTell}(y, x, \phi) \wedge \exists A \text{ s.t. } \langle A, y, x, \phi, 0\rangle \in \mathsf{AP} \Longrightarrow \sigma[4] = \mathbf{Argue}(y, x, 0, A) \quad \text{(G2)}$$

As the Server has given a rationale to the Client, the Client should reply to the Server. Formulas (G3) state that both agents should present acceptable arguments:

$$\begin{aligned} \sigma[i] = \mathbf{Argue}(y, x, 0, A) &\Longrightarrow A \in \mathrm{acceptable}(y) \\ \sigma[i] = \mathbf{Argue}(x, y, 1, A) &\Longrightarrow A \in \mathrm{acceptable}(x) \end{aligned} \quad \text{(G3)}$$

Agents are thoughtful according to [12]'s assertion attitudes. Now both Client and Server should present arguments in order to counter

the opponent. Let us first focus on the Client. When Client x evaluates the argument proposed by Server y it may face two cases whether it can reply or not to the Server:

The client can reply Whether the received argument is acceptable or not, Client x argues as long as it can. In such a configuration, Client x considers all of its *acceptable* arguments that defeat the received argument and presents them to the Server. Formula (G4) specifies this counter-argumentation as follows:

$$\sigma[i] = \mathbf{Argue}(y, x, 0, A) \Longrightarrow (\forall B \text{ s.t. } B \in \text{acceptable}(x) \wedge (B, A) \in \Re (\exists j(\sigma[j] = \mathbf{Argue}(x, y, 1, B))) \quad \text{(G4)}$$

The client cannot reply The received argument is acceptable w.r.t. set $\text{acceptable}(x)$: in that case the dialog is over if Client x can no longer present a counter-argument to Server y which is acceptable for x. Formula (G5) specifies in a formal way the closure of the dialog: the first line states that Client x has received an acceptable argument and second, third and fourth lines of formula (G5) state that x has presented all counter-arguments to y; more precisely the second line states for every argument presented against the permission sent by the Server, Client x has presented (line 4) all the possible counter-arguments (line 3).

$$\sigma[i] = \mathbf{Argue}(y, x, 0, A) \wedge A \in \text{acceptable}(x) \wedge (\forall B, \exists j < i(\sigma[j] = \mathbf{Argue}(y, x, 0, B) \Longrightarrow \forall C \in \text{acceptable}(x) \text{ s.t. } (C, B) \in \Re \exists k < i(\sigma[k] = \mathbf{Argue}(x, y, 1, C))) \Longrightarrow \sigma[i+1] = \mathbf{EndDialogue}(x, y) \quad \text{(G5)}$$

It follows that preferences play a key role: Client x presents counter-arguments that are better, w.r.t. its point of view, than the received argument. Notice that formula (G5) combined with formulas (S3) and (S4) entails that triggering the closure of the dialog ends the dialog.

Now, let us focus on the Server side. The formulas are similar to formulas (G5) and (G4): as long as the Server can present arguments to the Client to persuade it to not change the authorization, the Server presents the counter-arguments to the Client.

The Server can reply Formula (G6) is similar to formula (G4) and specifies that Server y presents all possible counter-argument to an argument presented by x:

$$\sigma[i] = \mathbf{Argue}(x, y, 1, A) \Longrightarrow (\forall B \text{ s.t. } B \in \text{acceptable}(y) \wedge (B, A) \in \Re (\exists j(\sigma[j] = \mathbf{Argue}(y, x, 0, B))) \quad \text{(G6)}$$

The Server cannot reply In order to write formula (G7) which specifies the end of the dialog, we first characterize the condition $\mathbf{\Psi}(i)$ which holds if at time i all arguments which appear in argued permissions have been sent (lines 1 and 2), all arguments presented by x has been countered (lines 3, 4 and 5).

$$\forall \langle A, y, x, \phi, 0\rangle \in \mathsf{AP} \; \exists j((j < i) \wedge (\sigma[j] = \mathbf{Argue}(y, x, 0, A))) \wedge \forall B, \exists k \; (k < i) \wedge (\sigma[k] = \mathbf{Argue}(x, y, 0, B) \Longrightarrow \forall C \in \text{acceptable}(y) \text{ s.t. } (C, B) \in \Re \; \exists l((l < i) \wedge (\sigma[l] = \mathbf{Argue}(y, x, 0, C)))) \quad (\mathbf{\Psi}(i))$$

Second, we state that if the argument received by y is acceptable and condition $\mathbf{\Psi}(i)$ hold (line 1) then it entails that Server y should evaluate the whole set of arguments sent by x so that it may change the permission and provide information ϕ, otherwise the dialog is closed:

$$\sigma[i] = \mathbf{Argue}(x, y, 1, A) \wedge \mathbf{\Psi}(i) \Longrightarrow (\sigma[i+1] = \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle) \vee \sigma[i+1] = \mathbf{EndDialogue}(y, x)) \quad \text{(G7)}$$

All these constraints enable to characterize the dialogs for negiotiating permissions.

Definition 9 (Permission Negotiation Dialog) *Let $D = \langle \text{control}, \text{perm}, \text{MPAF}, \mathsf{AP}, \sigma\rangle$ be a dialog. D is a* permission negotiation dialog *iff (i) all permissions are consistent, (ii) σ is finite and (iii) all formulas (S1)–(S4), (R1)–(R5) and (G1)–(G7) hold.*

A permission negotiation dialog does not specify how the Server may change the permission, it just specifies how arguments may be exchanged and how information may be provided. Its main characteristic is that the sequence of locutions is finite, i.e. the dialog always terminates and it terminates in a "sound way": information ϕ has been provided because (i) Client x has the permission or (ii) a negotiation has occurred and all arguments and counter-arguments related to argued permissions have been exchanged, or (iii) information ϕ has been provided and also in that case every argument related to the argued permissions have been exchanged.

Proposition 2 *Let $D = \langle \text{control}, \text{perm}, \text{MPAF}, \mathsf{AP}, \sigma\rangle$ be a permission negotiation dialog. The following properties hold:*

1. *termination: if $n = length(\sigma)$ then $\sigma[n] = \mathbf{EndDialogue}(x, y)$ or $\sigma[n] = \mathbf{EndDialogue}(y, x)$*
2. *justification of acceptance:*

$$\sigma[length(\sigma) - 1)] = \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle) \iff \text{perm}(y, x, \phi) = 1 \vee (\text{perm}(y, x, \phi) = 0 \wedge \mathbf{\Psi}(length(\sigma) - 1))$$

3. *justification of refusal:*

$$\sigma[length(\sigma) - 1)] \neq \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle) \Longrightarrow \mathbf{\Psi}(length(\sigma) - 1))$$

The final step is the evaluation of the Client's arguments by the Server in order to determine if permission has to be changed.

4.4 Changing the permission

Server y changes permission related to x and ϕ with respect to a set of rules which characterize principles of *cautiousness* (the server still has a reason not to change the permission) or *trustfulness* (the server has at least one reason to change the permission):

cautiousness One of the argument presented by the Server has not been defeated by the Client. In other words, the Server has at least one reason for not changing permission. Let (**C-Caut**) be a formula which represents this condition. (**C-Caut**) specifies that the Server has send an argument (line 1) so that the Client has not reply to this argument (lines 2 and 3) with an argument involved in an argued permission that prevents to give permission (line 3):

$$\begin{array}{l}\exists A, \exists i(\sigma[i] = \mathbf{Argue}(y, x, 0, A) \wedge \\ \quad \nexists B \in \text{acceptable}(y) \text{ s.t.}(B, A) \in \Re \wedge \\ \exists j(\sigma[j] = \mathbf{Argue}(x, y, 1, B) \wedge \langle B, y, x, 0, \phi\rangle \notin \mathsf{AP}))\end{array} \quad \textbf{(C-Caut)}$$

Formula (C1) specifies that if all arguments have been exchanged (condition ($\mathbf{\Psi}(length(\sigma) - 1)$ holds) and if condition (**C-Caut**) does not holds (i.e. Client x has countered all the arguments presented by Server y), then server y has to provide ϕ:

$$\begin{array}{l}\mathbf{\Psi}(length(\sigma) - 1)) \ \wedge \ \neg(\textbf{C-Caut}) \Longrightarrow \\ \quad (\sigma[length(\sigma - 1)] = \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle)\end{array} \quad \text{(C1)}$$

Once ϕ has been provided, the dialog is closed (see formula (R4)). It follows that permission has to be updated so that it reflects that Client x can access ϕ. Formula (C2) states that if ϕ has been provided w.r.t. the cautiousness principle then the permission is updated (perm' represents the new permission):

$$\begin{array}{l}\text{perm}'(y, x, \phi) := 1 \iff \\ \qquad \mathbf{\Psi}(length(\sigma) - 1) \ \wedge \ \neg(\textbf{C-Caut})\end{array} \quad \text{(C2)}$$

trustfulness One of the argument presented by the Client is acceptable for Server y. In other words, the Server has at least one reason to change permission. Formula (**C-Trust**) specifies the condition corresponding to this attitude as follows: lines 1 and 2 state that there exists at least one acceptable argument that is not against permission (according to Server point of view):

$$\begin{array}{l}\exists A, i(\sigma[i] = \mathbf{Argue}(x, y, 1, A) \wedge \\ A \in \text{acceptable}(y) \wedge \langle A, y, x, 0, \phi\rangle \notin \mathsf{AP}))\end{array} \quad \textbf{(C-Trust)}$$

Formula (C3) specifies that if condition (**C-Trust**) holds then information ϕ is provided.

$$\begin{array}{l}(\mathbf{\Psi}(length(\sigma) - 1) \ \wedge \ \textbf{(C-Trust)} \Longrightarrow \\ \quad (\sigma[length(\sigma - 1)] = \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle)\end{array} \quad \text{(C3)}$$

As previously, we now state permission change in a trustfulness context:

$$\begin{array}{l}\text{perm}'(y, x, \phi) := 1 \iff \\ \qquad (\mathbf{\Psi}(length(\sigma) - 1) \ \wedge \ \textbf{(C-Trust)}\end{array} \quad \text{(C4)}$$

Since the permission has changed, the set of argued permissions has also to be changed so that the new permission is consistent. That is every argument sent by the Client that is acceptable from the point of view of the Server has to be added to the list of argued permissions AP. Formula (C5) states that all argument received by y and acceptable by y extend the initial list of argued permissions.

$$\mathsf{AP}' := \begin{cases} \mathsf{AP} \text{ if } \nexists i(\sigma[i] = \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle)) \\ \mathsf{AP} \cup \{\langle A, y, x, \phi, 1\rangle \mid \\ \qquad \exists i(\sigma[i] = \mathbf{Argue}(x, y, 1, A) \wedge \\ \qquad A \in \text{acceptable}(y))\} \text{ otherwise} \end{cases} \quad \text{(C5)}$$

The first consequence entails by formulas (C1)–(C5) is that the updated permission is still consistent.

Proposition 3 *Let D be a permission negotiation dialog. Let* $\text{perm}'(y, x, \phi)$ *and* AP' *be the updated set of permissions defined w.r.t. formulas* (C1) *and* (C2)*, or formulas* (C3) *and* (C4)*; let* AP' *be the updated set of argued permissions calculated w.r.t. formula* (C5)*.* $\text{perm}'(y, x, \phi)$ *is* consistent *with respect to the set* AP'*.*

The second consequence is an entailment relation between the two policies: a permission that has been given w.r.t. the cautiousness principle entails that the permission should also have been given w.r.t. the trustfulness principle (but not vice-versa). This is due to the fact that whenever condition (**C-Caut**) does not hold, condition (**C-Trust**) holds.

Proposition 4 $\neg$(**C-Caut**) $\Longrightarrow$ (**C-Trust**)

Notice that trustfulness corresponds to the skeptical acceptance attitude of [12]; cautiousness is an acceptance attitude not taken into account by [12].

We conclude the section by assessing the principle of cautiousness and trustfulness whether Server y uses a skeptical or credulous notion of acceptability. As long as Server y uses a credulous acceptability permissions may change:

Proposition 5 *For all permission dialog D s.t.* $\text{acceptable}(y) = \mathsf{Cr}(y)$*, it holds that*

$$\begin{array}{l}\neg((\phi \in \text{control}(y) \wedge \text{perm}(y, x, \phi) = 0) \Longrightarrow \\ \qquad \nexists i(\sigma[i] = \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle))\end{array}$$

As long as Server y uses a skeptical notion of acceptability, Server y never changes its initial permissions and thus will never provide information when initial permission is equal to 0.

Proposition 6 *For all permission negotiation dialog D such that* $\text{acceptable}(y) = \mathsf{Sk}(y)$*, it holds that:*

$$\begin{array}{l}(\phi \in \text{control}(y) \wedge \text{perm}(y, x, \phi) = 0) \Longrightarrow \\ \qquad \nexists i(\sigma[i] = \mathbf{Tell}(y, x, \langle \text{content } \phi\rangle)\end{array}$$

It follows from the previous propositions that an agent which gives to the other agents the ability to negotiate permissions should not adopt a too restrictive notion of acceptability. That is, acceptability should be credulous based.

5. REVISITING THE INITIAL EXAMPLE

In this section we reformulate the dialog between the agent of Brussels Hospital and the London agent as a permission negotiation dialog $D = \langle \text{control}, \text{perm}, \text{MPAF}, \mathsf{AP}, \sigma\rangle$ such that control and perm are defined as in example 1, MPAF is defined as in example 2 and AP is defined as shown in example 3. Server y may change permissions w.r.t. trustfulness principle. We have the following sequence σ of locutions (relevant constraints that hold are mentioned on the right part of the locution):

Agent of Brussels Hospital: I would like to dialog with the agent of Robert's British doctor; I request Robert's health record (information ρ).

$$\sigma[1] = \mathbf{OpenDialogue}(b, l) \quad \text{(S1)}$$
$$\sigma[2] = \mathbf{Ask}(b, l, \rho) \quad \text{(R1)}$$

London agent: I cannot provide you Robert's health record because Robert has only given his British doctor limited consent to pass on his personal information (argument A_1).

$$\sigma[3] = \mathbf{DontTell}(l, b, \rho) \quad \text{(R5)}$$
$$\sigma[4] = \mathbf{Argue}(l, b, 0, A_1) \quad \text{(G2)}$$

Brussels agent: This record could possibly include information that could affect the treatment of Robert's heart failure. I request it, Robert's life may be at stake (argument A_2)!

$$\sigma[5] = \mathbf{Argue}(b, l, 1, A_2) \quad \text{(G1, G3, G4)}$$

London agent: I cannot divulge this information, because British law prohibits passing on information without the consent of the provider of the information (argument A_3).

$$\sigma[6] = \mathbf{Argue}(l, b, 0, A_3) \quad \text{(G3, G6)}$$

Brussels agent: EC law takes precedence over British law when it would be in the interests of the owner to divulge the information (argument A_4). You should allow me to access the record.

$$\sigma[7] = \mathbf{Argue}(b, l, 1, A_4) \quad \text{(G1, G3, G4)}$$

London agent: Only Robert could decide what would be in his interests (argument A_5).

$$\sigma[8] = \mathbf{Argue}(l, b, 0, A_5) \quad \text{(G1, G3, G6)}$$

Brussels agent: Robert's doctor owes a duty of care to Robert and, should he die, the doctor might be sued by his family, or the Brussels hospital, or both (argument A_6).

$$\sigma[9] = \mathbf{Argue}(b, l, 1, A_6) \quad \text{(G3, G4, G7)}$$

London agent: OK. I provide you the requested record: Robert's history of diabetes is...

$$\sigma[10] = \mathbf{Tell}(l, b, \langle \text{content } \rho \rangle) \quad \text{(C3, G7)}$$
$$\sigma[11] = \mathbf{EndDialogue}(b, l) \quad \text{(S3, S4, R4, C3)}$$

The following relevant conditional formulas also hold: (S2), (R2), (R3), (G1). According to the trustfulness principle (formula (C4)), London agent changes the permission, $\text{perm}'(l, b, \rho) = 1$, because there is an l-acceptable argument A_6 that makes condition (**C-Trust**) true. The set of argued permissions is also updated:

$$\mathsf{AP}' = \mathsf{AP} \cup \{\langle A_2, l, b, \rho, 1\rangle, \langle A_6, l, b, \rho, 1\rangle\} = \mathsf{AP}$$

6. CONCLUSION

In this paper, we have presented a formal framework for handling the negotiation of permissions. Our contribution is two fold: first we represent through an explicit link between arguments and permissions why agents accept of refuse to provide information. The agents can thus justify their behavior. Second, we exhibit a specific class of dialogs, *permission negotiation dialog*, which helps to characterize two policies of negotiation of permission (cautiousness and trustfulness). We have shown that enabling the negotiation of permission entails the evaluation of arguments in a credulous way. The proposed protocol has been shown in the context of multiple preferences argumentation framework, however this protocol of negotiation is sufficiently general so that it can be used with other argumentation frameworks.

Giving agents the ability to negotiate permissions enables the incorporation of some trust aspects [16]: first, trusted relationships between agents are explicitly represented in terms of permissions and second, the proposed protocol can be viewed as a protocol for gaining another agents' trust.

For future work we plan to extend our framework by embedding the proposed protocol in frameworks for negotiation process based on argumentation [15] where trust is a key issue such as in an electronic marketplace.

7. REFERENCES

[1] L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks . *International Journal of Automated Reasoning*, 29(2):125–169, 2002.

[2] L. Amgoud, S. Parsons, and L. Perrussel. An Argumentation Framework based on contextual Preferences . In *Proc. of FAPR'00, London*, pages 59–67, January 2000.

[3] H. Andreka, M. Ryan, and P. Schobbens. Operators and Laws for Combining Preference Relations. In R. Wieringa and R. Feenstra, editors, *Information Systems: Correctness and Reusability (Selected Papers)*. World Scientific Publishing Co., 1995.

[4] T. J. M. Bench-Capon. Specifying the interaction between information sources. In *Proc. of DEXA'98, Vienna, Austria*, volume 1460 of *LNCS*, pages 425–434. Springer, 1998.

[5] T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 13(3):429–448, 2003.

[6] G. Boella, J. Hulstijn, and L. van der Torre. Argument games for interactive access control. In *Proc. of WI 2005*, pages 751–754. IEEE CS, 2005.

[7] G. Boella, J. Hulstijn, and L. van der Torre. Argumentation for access control. In *Proc. AIIA'05*, pages 86–97, 2005.

[8] P. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and N-Person games. *Artificial Intelligence*, 77(32):321–357, 1995.

[9] FIPA. *FIPA, 'Agent communication language'*, FIPA 97 Specification, Foundation for Intelligent Physical Agents edition, 1997.

[10] A. Kakas and P. Moraitis. Adaptative agent negotiation via argumentation. *Proc. of AAMAS'06*, 2006.

[11] S. Parsons, P. McBurney, and M. Wooldridge. The mechanics of some formal inter-agent dialogues. In F. Dignum, editor, *Advances in Agent Communication*, volume 2922 of *LNCS*, pages 329–348. Springer-Verlag, 2003.

[12] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *J. Log.*

Comput., 13(3):347–376, 2003.

[13] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In *Proc. of JELIA'00*, volume 1919 of *LNAI*, pages 224–238. Springer-Verlag, 2000.

[14] H. Prakken and G. Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4:331–368, 1996.

[15] I. Rahwan, S. Ramchurn, N. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18:343–375, 2003.

[16] S. Ramchurn, T. Huynh, and N. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19:1–25, 2004.

[17] D. Walton and E. Krabbe. *Commitments in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, 1995.