Arguing for Gaining Access to Information

Sylvie Doutre[1], Peter McBurney[2], Laurent Perrussel[1] and Jean-Marc Thevenin[1]

[1]IRIT - Universite Toulouse I
2 rue du doyen Gabriel Marty – 31042 Toulouse Cedex 9 – France
{sylvie.doutre, laurent.perrussel, jean-marc.thevenin}@univ-tlse1.fr

[2]Dpt. of Computer Science - University of Liverpool
Liverpool L69 3BX – United Kingdom
p.j.mcburney@csc.liv.ac.uk

ABSTRACT

This paper presents a 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.

Categories and Subject Descriptors

I.2.11 [**Computing methodologies**]: Artificial intelligence—*Distributed Artificial Intelligence*

General Terms

Languages

Keywords

Dialogue, argumentation, permission

1. INTRODUCTION

In this paper, 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 [11, 10] 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 [10, 12, 13] for information-seeking as well as preference-based argumentation systems [2, 1, 3] have already been studied. 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 [4, 5] in the context of an argument-based dialog framework. None of them describes 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.

Based on this link, we propose a persuasion dialog protocol for access authorization. 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 is twofold: a 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. Section 2 presents how to link permissions and arguments. Section 3 describes the protocol that rules the dialog. We conclude the paper in section 4.

2. PERMISSIONS AND ARGUMENTS

In this section we describe the concepts of access rights and argumentation which help us to specify the persuasion 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...). 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).

AAMAS'07, May 14–18, 2007, Honolulu, Hawai'i, USA.

978-81-904262-7-5 (RPS)

2.1 Access to information

The *permission* a participant y has to access the content of information ϕ is denoted by a function $\mathrm{perm}(y, x, \phi)$: $\mathrm{perm}(y, x, \phi) = 1$ (respectively 0) stands for agent y can give (respectively cannot give) to agent x the content of information ϕ.

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 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.

2.2 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. [3] 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 [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. At this stage, we do not need either to enforce the usage of a specific notion of acceptability. Hence each agent evaluates the set of arguments w.r.t. 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 3). Formally, we obtain the following definition:

Definition 1 (MPAF) *A* Multiple preference-based argumentation framework *(MPAF) is a tuple* $\langle \mathsf{Arg}, \Re, \bigcup_{x \in \mathsf{Ag}} \geqslant_x, \mathrm{acceptable} \rangle$ *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, and* acceptable *is a function which maps agent ids to a subset of* Arg *which characterizes acceptability (*$\mathrm{acceptable} : \mathsf{Ag} \mapsto 2^{\mathsf{Arg}}$*);* $\mathrm{acceptable}(x)$ *stands for the acceptable set of arguments associated to agent* x*. Each set* $\mathrm{acceptable}(x)$ *is a subset of* Arg *defined w.r.t. the defeat relation* $\Re$ *and preference relation* $\geqslant_x$*.*

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. A usage of a specific acceptability notion based on [8, 1] is presented in [6].

2.3 Linking Arguments and Permissions

In order to connect permissions and arguments, the notion of an argued permission is defined. 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: Agent y has the argument A in favor ($\iota = 1$) or against ($\iota = 0$) giving permission to agent x to obtain information ϕ.

In fact, it is possible for agent 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. We claim that a permission defined by agent y about agent x and information 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. $\mathrm{acceptable}(y)$.

3. PERMISSION PERSUASION-DIALOG

In this section, we describe the protocol of a dialog system for information-seeking which requires permission to access the information.

Such a dialog involves two participants, a *Client* (requesting information), and a *Server* (controlling access to some information, which it may or may not agree to provide). 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. Client and Server may have disjoint knowledge bases. The knowledge base of the Server includes information about the access permissions which each Client has

The locutions of the dialogs should allow us to open and to close dialogs, to request some information, to provide the content of some requested information, to indicate that the content of some information cannot be provided, and to argue about the permission related to some requested information. These locutions may be based on FIPA standards [9].

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.

We present now in an informal way a protocol of dialog for information-seeking dialog with permissions (see [6] for a formal description).

Client x and Server y start with setting the dialogue. After this setting, Client x requests some information ϕ. Server y should provide ϕ if y controls ϕ and x has the authorization to access ϕ. If y actually provides information ϕ then the dialog should be closed by Client x or Server y. If y has no control over ϕ then it should close the dialog. Now if Server y controls ϕ but Client x does not have the authorization to access information ϕ (there exists in this case an argued permission $\langle A, y, x, \phi, 0 \rangle$ in the knowledge base of the Server), then y tells x that it cannot provide it with ϕ and y motivates its refusal by arguing A.

An argumentation-based negotiation stage starts: the Client will try to convince the Server to give it permission, whereas the Server

will try to convince the Client that it will not change the permission. To this end, Client and Server will present arguments acceptable w.r.t. their own point of view in order to counter the opponent (agents are thoughtful according to [10]'s assertion attitudes).

Let us first focus on the attitude of the Client when it receives an argument from the Server. Whether the received argument is acceptable or not, Client x argues as long as it can in order to get the permission. In such a configuration, x considers all its arguments of $\mathrm{acceptable}(x)$ that defeat the received argument and presents them as counter-arguments to Server y. These counter-arguments are better, w.r.t. its point of view, than the received argument. However, if the received argument is acceptable w.r.t. set $\mathrm{acceptable}(x)$, and if for every argument against the permission sent by the Server, x has presented all the possible counter-arguments, then the dialog is over. Client x has not been able to convince Server y to change the permission.

Now, let us focus on the Server side. The principle is similar to the Client side: as long as the Server can present arguments to the Client to persuade it that it will not change the authorization, the Server presents them to the Client. In other words, Server y considers all its arguments of $\mathrm{acceptable}(y)$ that defeat the received argument and presents them to Client x. If y cannot present any such argument, it presents every argument A that appears in an argued permission $\langle A, y, x, \phi, 0\rangle$. However, if all arguments which appear in argued permissions have already been sent, all arguments presented by the Client have already been countered, then the Server should evaluate the whole set of arguments sent by the Client. From this evaluation, either the Server decides to change the permission and provides the content of ϕ to the Client, or the Server decides not to change the permission. After this evaluation, the dialog is closed.

We propose two ways for the Server to evaluate the set of arguments sent by the Client. The Server may be *cautious*: if one of the arguments presented by the Server has not been defeated by the Client, then the Server still has a reason not to change the permission. To the contrary, Server y may be *trustful*: if one of the arguments presented by the Client belongs to $\mathrm{acceptable}(y)$, i.e. is acceptable for the Server, and if this argument does not appear in any argued permission against the permission, then the Server has a reason to change the permission. Consequently, the set of argued permissions is 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 is added to the list of argued permissions.

The rules previously described which characterize our protocol guarantee that a *permission persuasion dialog* is well-formed. At the end of such a dialogue, information ϕ has been provided because (i) Client x had the permission, or (ii) Client x did not had the permission, but a persuasion has occurred where all arguments and counter-arguments related to argued permissions have been exchanged, and the Server has finally decided to change the permission the Client had.

4. CONCLUSION

We have presented a framework for handling information-seeking with permissions. Our contribution is two fold: first we have represented 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 have exhibited a specific class of dialogs, *permission persuasion dialog*, which helps to characterize two policies of permission change (cautiousness and trustfulness). The proposed protocol has been shown in the context of multiple preferences argumentation framework, however this protocol of persuasion is sufficiently general so that it can be used with other argumentation frameworks. A formalisation of this protocol can be found in [6]. Such a protocol may be implemented by using the method proposed in [7].

As future work, we plan to extend the procotol to a family of protocols. That is, in this paper we focus on specific acceptability definitions; our aim is to consider the notions of conflict and acceptability at a more general level and to evaluate the impact on the proposed persuasion protocol. We also plan to refine the protocol in order to handle trust issues. That is if a client has been able to persuade a server to get permission to access some information, then this result may play the role of an argument in favor of the client for gaining access to some other information, i.e. the persuasion dialog may be viewed as a trust proof.

Acknowledgments

Peter McBurney is grateful for support from the EC project *Argumentation Service Platform with Integrated Components (ASPIC)* (IST-FP6-002307). Laurent Perrussel is grateful for support from the ANR project *Social trust analysis and formalization (ForTrust)*.

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

[4] 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.

[5] P. Dijkstra, F. Bex, H. Prakken, and C. De Vey Mestdagh. Towards a multi-agent system for regulated information exchange in crime investigations. *Artificial Intelligence and Law*, 13:133–151, 2005.

[6] S. Doutre, P. McBurney, L. Perrussel, and J.-M. Thevenin. Arguing for Gaining Access to Information. Research report IRIT/RR–2006-26–FR, IRIT, 2006.

[7] S. Doutre, P. McBurney, and M. Wooldridge. Law-governed linda as a semantics for agent dialogue protocols. In *AAMAS*, pages 1257–1258, 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] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *J. Log. Comput.*, 13(3):347–376, 2003.

[11] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *J. Log. Comput.*, 15(6):1009–1040, 2005.

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

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