Simon Parsons
Nicolas Maudet
Pavlos Moraitis
Iyad Rahwan (Eds.)

LNAI 4049

Argumentation in Multi-Agent Systems

Second International Workshop, ArgMAS 2005
Utrecht, The Netherlands, July 2005
Revised Selected and Invited Papers

Springer

Lecture Notes in Artificial Intelligence 4049

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Simon Parsons Nicolas Maudet
Pavlos Moraitis Iyad Rahwan (Eds.)

Argumentation in Multi-Agent Systems

Second International Workshop, ArgMAS 2005
Utrecht, The Netherlands, July 26, 2005
Revised Selected and Invited Papers

Springer

Volume Editors

Simon Parsons
Brooklyn College of the City University of New York
Department of Computer and Information Science
Brooklyn NY 11210, USA
E-mail: parsons@sci.brooklyn.cuny.edu

Nicolas Maudet
LAMSADE, Université Paris IX-Dauphine
75775 Paris Cedex 16, France
E-mail: maudet@lamsade.dauphine.fr

Pavlos Moraitis
Université René Descartes
Department of Mathematics and Computer Science
45 rue des Saints-Pères, 75270 Paris Cedex, France
E-mail: pavlos@math-info.univ-paris5.fr

Iyad Rahwan
The British University in Dubai, Institute of Informatics
P.O. Box 502216, Dubai, UAE
E-mail: irahwan@acm.org

Library of Congress Control Number: 2006928861

CR Subject Classification (1998): I.2.11, I.2, C.2.4, H.5.2-3

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-36355-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-36355-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11794578 06/3142 5 4 3 2 1 0

Preface

This volume is based on the Second Workshop on Argumentation in Multiagent Systems (ArgMAS). The workshop was held in conjunction with the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), at the University of Utrecht in Utrecht, The Netherlands, in July 2005. The workshop itself took place on July 26.

We are happy to report that the second workshop was just as popular and successful as its predecessor, held the previous summer in New York. We received 17 submissions, each of which was reviewed by at least three experts in the field, and ten of these papers were accepted for presentation at the workshop. Once again the workshop was graced by an invited lecture, this time by Frans van Eemeren of the University of Amsterdam, who talked on the subject of pragma-dialectics. The workshop attracted 31 participants, ensuring many questions for the speakers, and a healthy exchange of views during the discussion periods.

Following the practice established with the post-proceedings of the first ArgMAS workshop, we invited the presenters of all the accepted papers to prepare revised versions of their papers for this volume. In addition we approached authors of papers on directly related topics that had been presented in the AAMAS conference, and this gave us an additional seven papers. We further solicited one additional paper (details below) and were lucky enough that Prof. van Eemeren consented to send us a paper that covered the material of his invited talk.

That paper by van Eemeren, written in conjunction with his long-time collaborator Peter Houtlosser and entitled "The Case of Pragma-Dialectics" opens this volume and forms its first part. As its title suggests, the paper provides an overview of the pragma-dialectic view of argumentation — in brief, this is a view that seeks to combine a dialectical view of argumentative reasonableness with a pragmatic view of the verbal moves made in argumentative discourse.

The second part of the book, entitled "Foundations," contains four papers, all of which deal, in different ways, with some of the very basic issues in computerized models of argumentation. The first, "A Logic of Abstract Argumentation," by Boella, Hulstijn and van der Torre, picks up the problem of formalizing the kind of reasoning that one can achieve using a system of argumentation. The logic they derive makes it possible to express the properties of such an argumentation system — the system they focus on is the system proposed by Dung [2] and widely studied since — for example, one can express that if arguments a and b attack c, then either a attacks c or b attacks c. "On the Metalogic of Arguments," by Wooldridge, McBurney and Parsons is concerned with a closely related topic. This paper considers the formalization of argumentation at different levels of abstraction. Just as Boella et al. distinguish between constructing arguments and reasoning about the relationship between arguments, Wooldridge et al. are interested in capturing both this object-level and meta-level reasoning. However, Wooldridge et al. are interested less in formalizing a specific argumentation system in this way, and interested more in constructing a general framework for this kind of reasoning, making it possible, for instance, to reason about different logics of argumentation (in the sense of Boella et al.).

"Nested Argumentation and Its Application to Decision Making Over Actions" by Modgil then looks at the meta-level reasoning question from yet another angle. Modgil's work starts from the position of wanting to provide a general solution to the problem of resolving the difference between two arguments that each defeat the other, and he does this by allowing the representation of arguments for and against each of the two arguments being the stronger. These, of course, are meta-level arguments. Modgil further suggests that one can construct arguments about meta-level arguments, in the same kind of way as suggested by Wooldridge et al., and applies his approach to making decisions about actions. Finally in this section, "Testing Formal Dialectic" by Wells and Reed provides a description of Scenario$_{GC0}$, a framework for implementing computational dialectic systems, which the authors suggest can play the same kind of role in the development of computational dialectics as the fruit fly *Drosophila* plays in biology.

The third part of the book is concerned with negotiation. Negotiation was one of the first topics to be considered by researchers interested in using argumentation in multiagent systems, and, as a result, it is one of the areas of argumentation in multiagent systems in which the most progress has been made. The four papers in this part of the book report a number of new developments that extend the range of what is possible in a negotiation.

One of the main purposes of using argumentation in a negotiation is to introduce a measure of "persuasion" (which, of course, can be considered an entirely separate kind of argumentation) into a negotiation. One way that this persuasion can be achieved is through the use of threats and rewards — one agent can offer a reward in return for another agent accepting its proposal (a kind of side-payment in game theory terms), or can offer a negative reward for not accepting the proposal, a threat. "Formal Handling of Threats and Rewards in a Negotiation Dialogue" by Amgoud and Prade provides a model for dealing with these issues, showing how they can be seamlessly incorporated into the argumentation process.

When engaging in negotiation, an agent can aim to persuade using arguments that are based on logical force — arguments where the correctness of what is being said is paramount. Agents can also persuade by making use of arguments that are based on societal roles — arguments where who makes the argument is important as well. "Argument-Based Negotiation in a Social Context" by Karunatillake, Jennings, Rahwan and Norman is concerned with this social form of argumentation. In particular, they develop a representation of social influence, and show how it can be used to derive arguments that are then used in a negotiation.

Another important aspect of employing argument in pratical situations is knowing how best to argue — in other words how to use different patterns of locutions to best advance one's interests. This matter is addressed in "Practical Strategic Reasoning and Adpatation in Rational Argument-Based Negotiation" by Rovatsos, Rahwan, Fischer and Weiss. This paper describes a model that enables agents to learn how best to argue and shows that agent performance improves over time when this model is used. This demonstration of performance improvement is notable because, unlike much work on argumentation in multiagent systems, it is empirical and so involves a implementation of a dialogue system.

Finally, in "A Protocol for Arguing About Rejections in Negotiation," van Veenen and Prakken consider how agents might deal with proposals that are rejected. As they argue, rejected proposals are very informative — knowing why a proposal was rejected makes it possible to avoid making new proposals that are rejected for the same reason. Van Veenen and Prakken provide a protocol which allows rejected proposals to be questioned, and show how it can lead to shorter negotiations.

The fourth section, "Protocols," contains four papers on this topic, one that is currently of great interest within the computational dialetics community. The first paper in this section, "New Types of Inter-agent Dialogues" by Cogan, Parsons and McBurney, starts from the classification of dialogue types introduced by Walton and Krabbe [3], and concentrates on the pre-conditions that study identifies for the different types of dialogue that it examines. Cogan et al. show that considering different pre-conditions leads to a range of new types of dialogue with somewhat different aims from those examined by Walton and Krabbe. The paper enumerates some of these new kinds of dialogue, and suggests simple protocols that can achieve them, with the overarching idea that the point of identifying these new kinds of dialogue is to be able to combine them, together and along with exisiting kinds of dialogue, to create new forms of interaction between agents. Of course, in order to assemble new kinds of dialogue as novel combinations of existing dialogue types, one needs a mechanism for combining dialogues. This is exactly what is provided by Dimopoulos, Kakas and Moraitis in "Argumentation-Based Modelling of Embedded Agent Dialogues" — in their framework dialogues are combined by embedding one dialogue inside another. While such combinations have been suggested before, indeed they are suggested in [3], this is the first paper to seriously make an effort to formalize the process of combination in a way that considers the nature of the dialectical shift taken at such transitions. The result is a general framework for analyzing embedded dialogues, a framework in which one can identify whether certain embeddings are legal.

The idea of combining a number of different types of "atomic" dialogue into more complex dialogues is one way to develop a powerful theory of dialogues in which one constructs complex interactions from simple and well-understood components. Another way to permit complex interactions is to develop more complex protocols, protocols that can be instantiated in many different ways. This latter approach is the subject of the final two papers in this section of the book. In "Liberalizing Protocols for Argumentation in Multiagent Systems," Vreeswijk suggests one way to approach this objective, by proposing a framework for inquiry dialogue that is considerably more flexible than many existing protocols (though, of course, it pays for its flexibility in the sense that it is not possible to ensure that dialogues will terminate). Even more ambitious is "Protocol Synthesis with Dialogue Structure Theory" by McGinnins, Robertson and Walton. In this paper, the authors propose a language for defining protocols, and then use this to define a process by which protocols can by synthesized. Characterizing this process as a set of declarative transformation rules, as the authors do, makes it possible to equip agents with this set of rules and to have the agents define their own protocols for interaction.

The next section of the book focuses on deliberation and coalition formation. Amgoud's "An Argumentation-Based Model for Reasoning About Coalition Structures" exploits the structure of argumentation to handle coalitions. Argumentation frameworks provide a mechanism for resolving conflicting arguments, they identify which arguments are not attacked, or are only attacked by arguments that are themselves defeated. There is a similar requirement in coalition formation — it is necessary to determine which coalitions do not conflict with any other coalitions (in the sense of including the same agents) and which only conflict with coalitions that are ruled out by other conflicts. Amgoud, spotting this similarity, has devised a system which uses the machinery of argumentation to identify a conflict-free set of coalitions. The other paper in this section is "Argumentation-Based Multiagent Dialogues for Deliberation" by Tang and Parsons. This integrates a simple planning procedure with an argumentation-based dialogue that distributes plan construction across all the agents in the dialogue.

The final section of the book is concerned with consensus formation. Now, to some extent "all" work on argumentation is concerned with consensus formation, but in this section we find papers that are explicitly focussed on this topic. The process of establishing the kind of justified truth computed by argumentation systems — where we reach a consensus that something is true provided all attacks on it are defeated — is precisely the process that scientists go through when assessing the status of theories. The formalization of this process of scientific argumentation is the topic of Hunter's "Presentation of Arguments and Counterarguments for Tentative Scientific Knowledge," the paper we solicited that was not presented at an AAMAS event. In this paper Hunter shows how the system of argumentation he and Besnard have developed [1] can be used to capture conflicting pieces of scientific knowledge, and the relative strengths of those pieces of knowledge. This is followed by "Towards a Formal Framework for the Search of a Consensus Between Autonomous Agents" by Amgoud, Belabbes and Prade. This paper suggests a model that has much in common with the kinds of negotiation modes commonly used in multiagent systems. In a group setting one agent makes a proposal, and this is then discussed by the group until either it is accepted by all, or one agent rejects it. If the proposal is rejected, then another suggestion may be made and discussed in turn.

This use of argumentation in a dialogue, to allow the views of different agents to be integrated by having them put arguments for and against options, is the classical way to make use of the ability to argue. The same kind of process is used in the system described by "Argumentation-Supported Information Distribution in a Multiagent System for Knowledge Management" by Brena, Chesñevar and Aguirre. Brena et al. describe how they integrated argumentation into the JITIK system to control the distibution of information to users of the system. The dissemination process invokes argumentation to decide whether a specific piece of information should be delivered to a given user, and this is done if the information distribution agent and the personal agent for that user reach a consensus that the user wants to (or should) be a recipient of the information. The final paper in the book is "How Agents Alter Their Beliefs After an Argumentation-Based Dialogue" by Parsons and Sklar. This paper, as the name suggests, addresses the problem of how agents should revise their beliefs after they have completed a dialogue. The paper identifies a number of different aspects of this revision procedure, before showing that adopting the one that seems most promising will lead to agents that reach ever greater consensus the longer they continue to engage in dialogue.

We conclude this preface by extending our gratitude to the members of the Steering Committee, members of the Program Committee, and the auxiliary reviewers, who together helped make the ArgMAS workshop a success. We also thank the authors for their enthusiasm to submit papers to the workshop, and for revising their papers on time for inclusion in this book.

April 2006

Simon Parsons
Nicolas Maudet
Pavlos Moraitis
Iyad Rahwan

Program Chairs
ArgMAS 2005

References

1. Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.
2. 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.
3. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* State University of New York Press, Albany, NY, USA, 1995.

Organization

The ArgMAS workshops are run by the ArgMAS Steering Committee. Each year, a subset of the Steering Committee are appointed as Program Chairs, and marshall the Program Committee and organize the workshop proceedings for that year.

Program Chairs

Simon Parsons	City University of New York, USA
Nicolas Maudet	Université Paris Dauphine, France
Pavlos Moraitis	Université René Descartes-Paris 5, France
Iyad Rahwan	British University in Dubai, UAE (Fellow) University of Edinburgh, UK

ArgMAS Steering Committee

Antonis Kakas	University of Cyprus, Cyprus
Nicolas Maudet	Université Paris Dauphine, France
Peter McBurney	University of Liverpool, UK
Pavlos Moraitis	Université René Descartes-Paris 5, France
Simon Parsons	City University of New York, USA
Iyad Rahwan	British University in Dubai, UAE (Fellow) University of Edinburgh, UK
Chris Reed	University of Dundee, UK

Program Committee

Leila Amgoud	IRIT, France
Katie Atkinson	University of Liverpool, UK
Jamal Bentahar	Laval University, Canada
Frank Dignum	Utrecht University, Netherlands
Rogier van Eijk	Utrecht University, Netherlands
Anthony Hunter	University College, London, UK
Antonis Kakas	University of Cyprus, Cyprus
Nikos Karacapilidis	University of Patras, Greece
Nicolas Maudet	Université Paris Dauphine, France
Peter McBurney	University of Liverpool, UK
Pavlos Moraitis	Université René Descartes-Paris 5, France
Xavier Parent	King's College, UK
Simon Parsons	City University of New York, USA
Henry Prakken	Utrecht University, The Netherlands

Iyad Rahwan	British University in Dubai, UAE (Fellow) University of Edinburgh, UK
Chris Reed	University of Dundee, UK
Carles Sierra	IIIA, Spain
Katia Sycara	Carnegie Mellon University, USA
Francesca Toni	Imperial College, London, UK
Paolo Torroni	Università di Bologna, Italy
Bart Verheij	Maastricht University, The Netherlands
Gerard Vreeswijk	Utrecht University, The Netherlands
Mike Wooldridge	University of Liverpool, UK

Table of Contents

Part I: Invited Lecture

The Case of Pragma-Dialectics
Frans H. van Eemeren, Peter Houtlosser 1

Part II: Foundations

A Logic of Abstract Argumentation
Guido Boella, Joris Hulstijn, Leendert van der Torre 29

On the Meta-logic of Arguments
Michael Wooldridge, Peter McBurney, Simon Parsons 42

Nested Argumentation and Its Application to Decision Making over Actions
Sanjay Modgil 57

Testing Formal Dialectic
Simon Wells, Chris Reed 74

Part III: Negotiation

Formal Handling of Threats and Rewards in a Negotiation Dialogue
Leila Amgoud, Henri Prade 88

Argument-Based Negotiation in a Social Context
Nishan C. Karunatillake, Nicholas R. Jennings, Iyad Rahwan, Timothy J. Norman 104

Practical Strategic Reasoning and Adaptation in Rational Argument-Based Negotiation
Michael Rovatsos, Iyad Rahwan, Felix Fischer, Gerhard Weiss 122

A Protocol for Arguing About Rejections in Negotiation
Jelle van Veenen, Henry Prakken 138

Part IV: Protocols

New Types of Inter-agent Dialogues
Eva Cogan, Simon Parsons, Peter McBurney 154

Argumentation Based Modelling of Embedded Agent Dialogues
Yannis Dimopoulos, Antonis C. Kakas, Pavlos Moraitis 169

Liberalizing Protocols for Argumentation in Multi-agent Systems
Gerard A.W. Vreeswijk 182

Protocol Synthesis with Dialogue Structure Theory
Jarred McGinnis, David Robertson, Chris Walton 199

Part V: Deliberation and Coalition Formation

An Argumentation-Based Model for Reasoning About Coalition Structures
Leila Amgoud 217

Argumentation-Based Multi-agent Dialogues for Deliberation
Yuqing Tang, Simon Parsons 229

Part VI: Consensus Formation

Presentation of Arguments and Counterarguments for Tentative Scientific Knowledge
Anthony Hunter 245

Towards a Formal Framework for the Search of a Consensus Between Autonomous Agents
Leila Amgoud, Sihem Belabbes, Henri Prade 264

Argumentation-Supported Information Distribution in a Multiagent System for Knowledge Management
Ramón F. Brena, Carlos I. Chesñevar, José L. Aguirre 279

How Agents Alter Their Beliefs After an Argumentation-Based Dialogue
Simon Parsons, Elizabeth Sklar 297

Author Index 313

The Case of Pragma-Dialectics*

Frans H. van Eemeren and Peter Houtlosser

University of Amsterdam

Abstract. The pragma-dialectical approach to argumentation aims to provide a sound integration of both dialectics — the study of critical exchanges — and pragmatics — the study of language use in actual communication. Pragma dialectics thus combines a dialectical view of argumentative reasonableness with a pragmatic view of the verbal moves made in argumentative discourse. This paper provides an overview of the current state of the pragma-dialectical approach, insofar as this can be done adequately in a single paper, and provides many pointers to the full range of work in this area.

1 The Pragma-Dialectical Approach to Argumentation

In the pragma-dialectical approach to argumentation the term argumentation is used to refer to a process ("I am still in the middle of my argumentation") as well as to its result ("Let's examine what her argumentation amounts to"). Characteristically, argumentation is then studied from a communicative perspective. This communication, which can be oral or written, will generally take place by verbal means, but non-verbal elements (such as gestures and images) may also play a part. In pragma-dialectics, argumentation is viewed as aimed at resolving a difference of opinion by critically testing the acceptability of the standpoints at issue. Thus perceived, the study of argumentation does not only have a descriptive dimension that pertains to the way in which argumentation is conducted in communicative practice but also a normative dimension pertaining to the norms of reasonableness that are employed when argumentation is judged for its quality and possible flaws are detected.

Logicians, whether they are in favor of a formal or an informal approach, tend to concentrate on the problems involved in the regimentation of reasoning. Social scientists and linguists, particularly discourse and conversation analysts, generally focus on empirical observation of argumentative discourse and its effects.[1] In the pragma-dialectical view, however, these two approaches must be closely interwoven. Both the limitations of non-empirical regimentation and those of non-critical observation need to be systematically transcended. Pragma-dialecticians make it their business to clarify how the gap between normative and descriptive insight can be methodically bridged. This objective can only be achieved with the help of a coherent research program in which a systematic connection — *a trait d'union* — is created between well-considered regimentation and careful observation.

[*] This article, which gives an overview of the pragma-dialectical approach, is for a large part based on [15] and [21]. A textbook version is in preparation.

[1] For protagonists of a purely normative or a purely descriptive approach, see [4] and [58, 59], respectively.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 1–28, 2006.

Following a classical tradition, the study of the regimentation of critical exchanges is called *dialectics*. The study of language use in actual communication, which belonged in the past largely to the domain of rhetoric, is nowadays generally called *pragmatics*. Hence the choice of the name *pragma-dialectics* for the approach to argumentation that aims for a sound integration of insight from these two studies. Pragma-dialectics combines a dialectical view of argumentative reasonableness with a pragmatic view of the verbal moves made in argumentative discourse.[2]

2 The Five Components of the Pragma-Dialectical Research Program

Because the pragma-dialectical research program is designed to achieve a well-considered integration of normative and descriptive insight, it is on the one hand aimed at developing a philosophical ideal of critical reasonableness and, grounded in this ideal, a theoretical model for acceptable argumentative discourse in a critical discussion. On the other hand, argumentative reality is investigated empirically to acquire an accurate description of the actual processes of argumentative discourse and the factors influencing their outcome. Starting from the results achieved in these two enterprises, the conceptual tools are developed to analyze argumentative reality in light of the critical ideal of reasonableness. Then the individual and the procedural problems of the practical analysis, evaluation and production of argumentative discourse — the alpha and omega of the study of argumentation — can be tackled methodically. The research program thus includes a philosophical, a theoretical, an empirical, an analytical, and a practical component.[3]

The fundamental question in the philosophical component is what it means to be reasonable in argumentation. As it happens, the conceptions of reasonableness entertained by argumentation scholars diverge from the outset, leading to quite different outlooks on what acceptable arguments are considered to be. Dialecticians maintain a critical outlook. For them, reasonableness does not solely depend on inter-subjective agreement on the norms, as many rhetoricians think, but also on whether these norms are conducive to the goal of resolving a difference of opinion by way of a critical discussion. Because the ideal of reasonableness is linked to the methodic conduct of a critical discussion, the dialectical philosophy of reasonableness is *critical-rationalist*.

In the theoretical component the philosophical ideal of reasonableness is given a shape by designing a model of what is involved in acting reasonably in argumentative discourse.

[2] The dialectical conception of reasonableness is inspired by critical rationalists and analytic philosophers, such as Popper [49, 50], Albert [1], and Naess [45], and by formal dialecticians and logicians, such as Hamblin [29], Lorenzen and Lorenz [44], and Barth and Krabbe [3]. The pragmatic conception of argumentative discourse as consisting of making regulated communicative moves is rooted in Austin [2] and Searle's [51, 52] ordinary language philosophy, Grice's [27] theory of rationality in discourse, and other studies of communication by discourse and conversation analysts. It is in the first place the combination of dialectical and pragmatic insight that distinguishes pragma-dialectics from 'formal dialectic' as developed by Barth and Krabbe [3] that incorporates dialectical insight in a formal (logical) approach.

[3] For a more elaborate explanation of the research program, see [15, ch. 2].

A theoretical model, like the Toulmin [53] model, aims at getting an adequate grasp of argumentative discourse by specifying modes of arguing and indicating when they are acceptable. The model serves as a conceptual and terminological framework that can fulfill heuristic, analytical, and critical functions in dealing with argumentative discourse. A dialectical model provides rules that specify which moves can contribute to resolving a difference of opinion in the various stages of a critical discussion. If this discussion is viewed, pragmatically, as an interaction of speech acts, the model is *pragma-dialectical*.

In the empirical component insight is sought after in the actual processes of producing, interpreting, and assessing argumentative discourse and the factors that influence their outcome. Such insight is acquired by carrying out qualitative and quantitative research. Qualitative research consists primarily in making observations by means of introspection and case studies, the (sometimes connected) quantitative research consists in experimental and statistical studies. In pragma-dialectical empirical research the emphasis is on explaining how various factors and processes play a role in argumentative reality in resolving a difference of opinion. The interest centers on the aspects of argumentative discourse that affect its *cogency*.[4]

In the analytical component a pragma-dialectical reconstruction of argumentative discourse is made to achieve an 'analytic overview' of the discourse that constitutes a proper point of departure for a critical evaluation. In argumentative discourse things are not only not always immediately obvious, they even may be different from what they seem. Sometimes a more or less complicated reconstruction is needed of what is said before an analysis can be justifiably made. Such a reconstruction takes always places from a perspective that focuses on specific aspects of the discourse, highlighting certain elements while ignoring others. A comparison with a stereotypical Freudian analyst may offer some clarification. Our Freudian analyst examines what is said from a psychological perspective, making use of the analytical tools provided by a particular theoretical background. She is, for instance, interested in mother fixation, signs of inferiority complexes and the likes. It goes without saying that she can only come to an analysis by examining carefully what has actually been said, or conspicuously left out, by her client. She cannot diagnose him as suffering from mother fixation right after the introduction. Neither can she do this on the sole ground that he has been singing the praise of his mother at every session. Nevertheless, after a careful reconstruction of certain things he said or implied, she might be justified to attribute a mother fixation to him because adding up a series of observations may warrant this analysis. Similarly, in a pragma-dialectical analysis of argumentative discourse a reconstruction of the discourse is carried out that starts from the theoretical model of a critical discussion, with its various stages and division of speech acts, and takes all knowledge gained by empirical investigation methodically into account. In pragma-dialectics, the central question in the analytical component is how argumentative discourse can be reconstructed in such a way that all those, and only those, aspects are highlighted that are relevant to resolving a difference of opinion on the merits. The resulting analysis can therefore be characterized as *resolution-oriented*.

[4] For pragma-dialectical research into the identification of argumentation that is cogency-centered, see, e.g., [26]. Cf. for experiments concentrating on deductive reasoning [46] and [40].

Finally, in the practical component of the research program methods are developed for improving individual skills and specific argumentative procedures. Argumentative competence involves a complex of dispositions whose mastery is gradual and relative to specific communicative situations. This means that argumentative skills can only be measured adequately by applying standards relating to particular types of argumentative endeavors. To improve argumentative practice by way of education or otherwise, argumentation must therefore be studied in a diversity of institutionalized and non-institutionalized contexts, ranging from the formal context of law to the informal context of a conversation with friends. In the practical component, pragma-dialecticians put their philosophical, theoretical, analytical and empirical insight to good use in developing methods for improving argumentative practice while taking account of circumstantial diversity. Because of its emphasis on furthering an awareness of the prerequisites for resolving differences of opinion and stimulating a discussion-minded attitude, the pragma-dialectical approach to the improvement of argumentative practice can be characterized as *reflection-minded.*

3 Four Meta-Theoretical Premises Serving as Methodological Principles

In carrying out the pragma-dialectical research program, argumentation is approached with four meta-theoretical premises. These basic premises serve as methodological principles in their concern about how one ought to set about studying argumentation. They constitute a basis for integrating the descriptive dimension of the study of argumentation with the normative dimension.[5]

First, *functionalization.* Argumentation is usually studied as a structure of logical derivations, psychological attitudes or epistemic beliefs rather than a complex of verbal (and non-verbal) acts that have a specific communicative function in a context of disagreement. As a result, argumentation is often described in purely structural terms, not only in formal and informal logical approaches, but also in studies of fallacies and practical argumentation. Such structural descriptions tend to ignore the functional rationale of the design of the discourse. The general function of argumentation is managing disagreement. It arises in response to, or anticipation of, a difference of opinion, and the lines of justification chosen in argumentative discourse are contrived to resolving the difference. The study of argumentation should therefore concentrate on the function of argumentation in the verbal management of disagreement.

Second, *socialization.* Especially in approaches concentrating on reasoning, argumentation is usually seen as the expression of individual thought processes. The central question then becomes assessing whether and how the elements that constitute the reasoning hold together in order to validate the arguer's position. But argumentation does not consist in a single individual privately drawing a conclusion and it is not put forward in a social vacuum. It is part of a communication process whereby two or more individuals who have a difference of opinion try to arrive at an agreement. Argumentation presupposes two distinguishable discussion roles, that of a protagonist

[5] The metatheoretical premises are for the first time explained in [10].

of a standpoint and that of a — real or projected — antagonist. It reflects the collaborative way in which the protagonist in the fundamentally dialogical interaction responds to the questions, doubts, objections, and counterclaims of the antagonist. This is why argumentation should be put in the social context of a process of joint problem solving.

Third, *externalization*. To find out whether or not their opinions will be accepted, people put their standpoints by way of their argumentation to certification, submitting their reasoning to public scrutiny. Channeled by a system of public commitment and accountability, the beliefs, inferences and interpretations that underlie argumentation are expressed or projected in the discourse. Whereas the motives people have for holding a position might be different from the grounds they offer and accept in its defense, what they can be held committed to is the position they have expressed in the discourse, whether directly or indirectly.[6] For that reason, all efforts to reduce argumentation to a structure of attitudes and beliefs or a chain of reasoning are inadequate. Rather than speculating about the psychological dispositions of the people involved in argumentation, the study of argumentation should concentrate on their commitments as externalized in, or externalizable from, the way in which they have expressed themselves in a certain context and on the consequences these commitments have for the process of argumentation.[7]

Fourth, *dialectification*. Argumentation is appropriate for resolving a difference of opinion only if it is capable of accommodating the relevant critical reactions of the antagonist. Discourse and conversation analysts generally restrict themselves to describing argumentation as it occurs, without any regard for how it ought to occur if it is to be appropriate for resolving a difference of opinion. A theory of argumentation, however, must be attentive to critical standards for assessing a discussion aimed at resolving a difference of opinion. This can be achieved by considering argumentation to be subjected to a dialectical procedure for resolving differences of opinion that is problem-valid as well as intersubjectively valid. The problem-validity of a procedure for conducting a critical discussion depends on how efficient and efficacious it is in furthering the resolution of a difference of opinion and excluding fallacious moves; its inter-subjective validity depends on its acceptability to the parties involved.[8] To transcend a merely

[6] This does not mean that it is not important to find out to what extent and in which ways 'internal' reasoning and 'external' argumentation diverge, but this research can only be carried out methodically if the two concepts are kept separate.

[7] The principle of externalization is at odds with those rhetorical approaches that explain the effectiveness of argumentation by referring, without any further ado, to the presumed psychological states of arguers and their audiences.

[8] This terminology was introduced by Barth and Krabbe [3, pp. 21–22]. In their usage, a discussion procedure that fulfills these requirements may claim 'problem solving validity' and '(semi-)conventional validity'. Semi-conventional validity amounts to intersubjective validity. A series of empirical experiments were carried out to test the inter-subjective acceptability of the critical normativity encapsulated in the pragma-dialectical rules [9, 16, 17]. The results provide insight in ordinary language users' reasonableness conceptions, their consistency, and the social, cultural and other differences between them. They also provide an empirical basis for developing pedagogically adequate textbooks. O'Keefe [47] makes clear that a normative ideal, in this case argumentative explicitness, may also be persuasively effective.

descriptive stance, argumentative discourse should therefore be viewed from the perspective of a dialectical procedure for critical discussion that is valid in both respects.[9]

4 The Model of a Critical Discussion

In pragma-dialectics, externalization, socialization, functionalization, and dialectification of argumentation is realized by systematically combining pragmatic and dialectical insight. Functionalization is achieved by making use of the fact that argumentative discourse occurs through — and in response to — speech act performances. Identifying the complex speech act of argumentation and the other speech acts involved in resolving a difference of opinion makes it possible to specify the relevant 'identity conditions' and 'correctness conditions' of these speech acts.[10] In this way, for instance, a specification can be given of what is 'at stake' in advancing a certain 'standpoint', so that it becomes clear what the 'disagreement space' is and how the argumentative discourse is organized around this context of disagreement.[11] Socialization is achieved by identifying who exactly take on the discussion roles of protagonist and antagonist in the collaborative context of argumentative discourse. By extending the speech act perspective to the level of interaction, it can be shown in which ways positions and argumentation in support of positions are developed. Externalization is achieved by identifying the specific commitments that are created by the speech acts performed in a context of argumentative interaction.[12] Rather than being treated as internal states of mind, in a speech act perspective notions such as 'disagreement' and 'acceptance' can be defined in terms of discursive activities. 'Acceptance', for instance, can be externalized as giving a preferred response to an arguable act. Finally, dialectification is achieved by regimenting the exchange of speech acts aimed at resolving a difference of opinion in a model of a perfect critical discussion. Such an idealized modeling of the systematic exchanges of resolution-oriented verbal moves, defines the nature and distribution of the speech acts that play a part in resolving a difference of opinion.

The pragma-dialectical model of a critical discussion is a theoretically motivated system for resolution-oriented discourse. Although the model is an abstraction, rather than merely serving as a Utopian ideal, it should provide people who wish to resolve

[9] According to Wenzel [57, p. 84], a dialectical approach views argumentation as a systematic management of discourse for the purpose of achieving critical decisions. Its purpose is to establish how discussions should be carried out systematically in order to critically test standpoints. To avoid the dangers of absolutism (or skepticism) and relativism, a dialectical procedure for critical discussion that agrees with a 'critical' philosophy of reasonableness incorporates both the product-oriented and process-oriented approaches to argumentation based on the 'geometrical' (logical) and the 'anthropological' (rhetorical) philosophies of reasonableness. For these philosophies, see Toulmin [54].

[10] For a definition of argumentation as a complex speech act, see van Eemeren and Grootendorst [10, pp. 39–46], [13, pp. 30–33]; for the speech act of advancing a standpoint, see Houtlosser [32]; for the distinction between identity conditions and correctness conditions, see van Eemeren and Grootendorst [13, pp. 30–31].

[11] The term disagreement space was introduced in [33, p. 261].

[12] A kindred approach to argumentation in which commitments as well as other basic concepts of pragma-dialectics also play a crucial role is [56].

their differences by means of argumentative discourse with vital guidance for their conduct.[13] The model must be constructed in such a way that it can serve not only as a paradigm for systematic reflection upon one's active oral and written participation in argumentative discourse, but also, and even more so, as a point of reference in analyzing and evaluating argumentative discourse. In addition, it can be a standard for guiding the methodical improvement of argumentative practice.

When developing a model of a critical discussion, one first needs to realize that resolving a difference of opinion is not identical with settling a dispute — the point of settling a dispute merely being that a difference of opinion is brought to an end.[14] A difference of opinion is resolved only if the parties involved have reached agreement on whether or not the disputed opinion is acceptable. This means that one party has either been convinced by the other party's argumentation, or the other party, realizing that its arguments cannot stand up to the first party's criticisms, withdraws the standpoint.[15] This is why a dialectical procedure designed for methodically resolving differences of opinion is a crucial part of the pragma-dialectical model of a critical discussion.

In a critical discussion, the parties attempt to reach agreement about the acceptability of the standpoints at issue by finding out whether or not these standpoints are defensible against doubt or criticism. The dialectical procedure for conducting a critical discussion is in the first place a method for exploring the acceptability of standpoints. In a critical discussion, the protagonist and the antagonist of a particular standpoint try to establish whether this standpoint, given the point of departure acknowledged by the parties, is tenable in the light of critical responses.[16] To be able to achieve this purpose, the dialectical procedure for conducting a critical discussion should not deal only with inference relations between premises and conclusions (or 'concessions' and 'standpoints'), but cover all speech acts that play a part in examining the acceptability of standpoints. In pragma-dialectics, the concept of a critical discussion is therefore given shape in a model that specifies all the various stages the resolution process has to pass and all the types of speech acts that are instrumental in any of these stages.

5 Stages in Resolving a Difference of Opinion

The stages that are to be distinguished analytically in the process of resolving a difference of opinion correspond with the different phases an argumentative discourse must

[13] In spite of their different philosophical roots, Habermas's [28] ideal speech situation and the ideal model of a critical discussion are in some respects similar. In pragma-dialectics, however, instead of viewing communication as aimed at achieving consensus, intellectual doubt and criticism are seen as the driving forces of progress, and should lead to a continual flux of opinions.

[14] A dispute may also be settled by relying on the arbitration of a third party, such as an umpire, a referee or a judge, but then it has not really been resolved.

[15] A critical discussion reflects the Socratic dialectic ideal of rational testing of any conviction, not only of statements of a factual kind but also of normative standpoints and value judgments [1]. Starting from the fallibility of all human standpoints, critical rationalists elevate the methodological concept of critical testing to the guiding principle of problem-solving.

[16] In accordance with their critical rationalist philosophy, dialecticians place great emphasis on the consequence of the fact that a proposition and its negation cannot both be acceptable at the same time. The testing of standpoints is thus equated with the detection of inconsistencies [1, p. 44].

pass through, albeit not necessarily explicitly, in order to resolve a difference of opinion. Ideally, the discussion starts with a confrontation stage, in which a difference of opinion manifests itself through an opposition between a standpoint and non-acceptance of this standpoint. In real argumentative discourse, this stage corresponds with those parts of the discourse where it becomes clear that there is an opinion that coincides with — real or projected — doubt or contradiction, so that a (potential) disagreement arises. If there is no confrontation of views, then there is no need for critical discussion.

In the opening stage of a critical discussion, the initial commitments — procedural, substantive, or otherwise — of the participants in the dispute are identified and it is decided who will act as protagonist or antagonist. A protagonist undertakes the obligation to defend the standpoint at issue while an antagonist assumes the obligation to respond critically to this standpoint and the protagonist's defense.[17] This stage is manifest in those parts of the discourse where the parties express themselves as such and explore whether there is sufficient common ground. If there is no such opening for an exchange of views, having a critical discussion does not make sense.

In the argumentation stage a protagonist of a standpoint methodically defends this standpoint against critical responses of the antagonist. If the antagonist is not yet wholly convinced of all or part of the protagonist's argumentation, he or she elicits new argumentation from the protagonist, and so on. As a consequence, the protagonist's argumentation can vary from very simple to extremely complex, and the 'argumentation structure' of the one argumentative discourse may be much more complicated than that of the next.[18] The argumentation stage is gone through in those parts of the discourse in which one party adduces arguments to overcome the other party's doubts, and the other party reacts. If there is no argumentation and no critical appraisal of argumentation, there is no critical discussion and the difference of opinion will remain unresolved.

In the concluding stage the protagonist and the antagonist of a standpoint determine whether the protagonist's standpoint has been successfully defended against the critical responses of the antagonist. If the protagonist's standpoint has to be withdrawn, the dispute is resolved in favor of the antagonist; if the antagonist's doubts have to be retracted, it is resolved in favor of the protagonist. If the parties do not draw any conclusions about the result of their attempts to resolve a difference of opinion, no successful completion of a discussion has been reached. A completion of a critical discussion that is successful, however, does not preclude that the same parties embark upon a new discussion. This new discussion may relate to a completely different difference of opinion, but also to an altered version of the same difference, while the discussion roles of the participants may switch or remain the same. In any event, the new discussion that then begins must again go through the same stages — from confrontation to conclusion.

[17] The role of antagonist may coincide with that of protagonist of another — contrary — standpoint, but this need not be so. Expressing doubt regarding the acceptability of a standpoint is not necessarily equivalent with adopting a contrary standpoint of one's own. If the latter is the case, the difference of opinion is no longer 'non-mixed', but 'mixed' [13, pp. 13–25].

[18] For an analysis of how different types of argumentation structures can come into being, see Snoeck Henkemans [30].

6 Distribution of Speech Acts in a Critical Discussion

Which speech acts can contribute in the various stages of a critical discussion to the resolution of a difference of opinion? To answer this question, it is useful to distinguish between five basic types of speech acts that can be performed in argumentative discourse.[19] When pointing out the roles that several types of speech acts can fulfill in resolving a difference of opinion it is important to emphasize, right from the start, that in argumentative discourse a great many speech acts are performed implicitly or indirectly, so that a certain role in a critical discussion may be fulfilled by different speech acts. We shall return to this subject when we explain analysis as reconstruction.

A first type of speech acts consists of the assertives. The prototype is an assertion by which the speaker or writer guarantees the truth of the proposition being expressed: "I assert that Chamberlain and Roosevelt never met." Assertives, however, not only relate to the truth of propositions but also to their acceptability in a wider sense ("Baudelaire is the best French poet"). Assertives are, for instance, denying and conceding. In a critical discussion, assertives can express a standpoint at issue, be part of argumentation in defense of a standpoint, and establish a conclusion ("I can maintain my standpoint"). The commitment to a proposition expressed in an assertive may vary from strong, as in the case of an assertion or statement, to fairly weak, as in a supposition.

A second type of speech acts consists of the directives. The prototype is an order, which requires a special position of the speaker or writer vis-à-vis the listener or reader: "Come to my room" can only be an order if the speaker is in a position of authority, otherwise it is a request or an invitation. A question is a special form of request: it is a request for a verbal act — the answer. Other examples of directives are forbidding, recommending, and challenging. Not all directives can play a role in a critical discussion: their role consists in challenging the party that has advanced a standpoint to defend this standpoint or in requesting argumentation to support a standpoint or (part of an) argumentation. A critical discussion does not allow for unilateral orders and prohibitions.

A third type of speech acts consists of the commissives. These are speech acts by means of which the speaker or writer undertakes a commitment vis-à-vis the listener or reader to do something or refrain from doing something. The prototype is a promise: "I promise you I won't tell your father". The speaker or writer can also undertake commitments about which the listener or reader may be less enthusiastic: "I guarantee that if you walk out now you will never set foot in this house again." Other commissives are accepting, rejecting, undertaking, and agreeing. In a critical discussion, commissives fulfill a series of roles: (not) accepting a standpoint, (not) accepting argumentation, accepting the challenge to defend a standpoint, deciding to start a discussion, agreeing to take on the role of protagonist or antagonist, agreeing on the rules of discussion, and deciding to begin a new discussion. Some of the required commissives, such as agreeing on the rules, can only be performed in cooperation with the other party.

A fourth type of speech acts consists of the expressives. By means of such speech acts the speaker or writer expresses his or her feelings about something by thanking

[19] This typology is largely based on Searle [52, pp. 1–29].

someone, revealing disappointment, and so on. There is no single prototypical expressive. Joy is expressed in “I’m glad to see you’re quite well again” and hope is echoed by “I wish I could find such a nice girl friend”. Other expressives include commiserating, regretting, and greeting. In a critical discussion, expressives, as such, play no constitutive role.[20]

A fifth type of speech acts consists of the declaratives. The performance of these speech acts creates a reality by calling a particular state of affairs into being. If an employer addresses an employee with the words “You’re fired”, he is not just describing a state of affairs but the words actually make a reality. Declaratives are usually bound to a specific institutionalized context in which certain people are qualified to perform a certain declarative: “I open the meeting” only works if you are the chair. [21]

7 Analysis as Reconstruction

For various reasons, argumentative reality does not always resemble the ideal of a critical discussion. According to the ideal model, for example, in the confrontation stage antagonists of a standpoint must state their doubts clearly and unambiguously, but in practice doing so can be ‘face-threatening’ for both parties so that they have to operate circumspectly.[22] Analyzing argumentative discourse pragma-dialectically amounts to interpreting the discourse from the theoretical perspective of a critical discussion. Such an analysis is pragmatic in viewing the discourse as essentially an exchange of speech acts; and dialectical in viewing this exchange as a methodical attempt to resolve a difference of opinion. A pragma-dialectical analysis is aimed at reconstructing all those, and only those, speech acts that play a potential part in bringing a difference of opinion to a conclusion. In accomplishing a systematic analysis the ideal model of a critical discussion is a valuable tool. By pointing out which speech acts are relevant in the various stages of the resolution process the model has the heuristic function of indicating which speech acts need to be considered in the reconstruction.

Van Eemeren, Grootendorst, Jackson and Jacobs further developed the analytical component of pragma-dialectics in *Reconstructing Argumentative Discourse* [19]. They emphasize that it is crucial that the reconstructions proposed in the analysis are indeed justified. The reconstructions should be faithful to the commitments that may be ascribed to the participants on the basis of their contributions to the discourse.[23] In order not to ‘over-interpret’ what seems implicit in the discourse, the analyst must be sensitive

[20] This does not mean that they cannot affect the course of the resolution process: sighing that you are unhappy with the discussion, expresses your emotions, which distracts the attention from the resolution process.

[21] Due to their dependence on the authority of the speaker or writer in a certain institutional context, declaratives can sometimes lead to a settlement of a dispute.

[22] Expressing doubt may also create a potential violation of the ‘preference for agreement’ that governs normal conversation. See Heritage [31, pp.265–280], Levinson [43, pp.332–336]), and van Eemeren, Grootendorst, Jackson, and Jacobs [19, ch. 3].

[23] Only in exceptional cases, when interpreting a move as a potential contribution to the resolution process is the only charitable option left, an unsupported reconstruction may be warranted ‘for reason’s sake’. See van Eemeren and Grootendorst [15, ch. 5].

to the rules of language use,[24] the details of the presentation, and the contextual constraints inherent in the speech event concerned. So as to go beyond a naïve reading of the discourse, empirical insight concerning the way in which oral and written discourse are conducted will be beneficial.[25] The analyst's intuitions can thus be augmented by the results of (qualitative and quantitative) empirical research.[26]

In practice, the first question always is whether, and to what extent, an oral or written discourse is indeed argumentative. Sometimes the discourse, or part of it, is explicitly presented as argumentative.[27] Sometimes it is not, even though it clearly has an argumentative function. There may also be cases in which the discourse is clearly not argumentative — or at least not primarily. The most decisive demarcation criterion is whether or not argumentation is advanced, so that the discourse is, at least partially, aimed at overcoming the addressee's — real or projected — doubt regarding a standpoint. A discourse can only be justifiably analyzed as argumentative, albeit not necessarily in toto, if, whether directly or indirectly, the complex speech act of argumentation is performed.

8 An Analytic Overview of Argumentative Discourse

In order to make it possible to evaluate argumentative discourse in a responsible way, an analytic overview is required of all elements in the discourse that are relevant to resolving a difference of opinion. Achieving such an overview is therefore the aim of the analysis. In an analytic overview the following points need to be attended to:

1. the issues that are at stake in the difference of opinion;
2. the positions the parties adopt and their procedural and material starting points;
3. the arguments explicitly or implicitly advanced by the parties;
4. the argumentation structure of the complex of arguments advanced in defense of a standpoint;
5. the argument schemes used in the individual arguments to justify a standpoint.

The terms and concepts referred to in the components of an analytic overview, such as unexpressed premise, argumentation structure and argument scheme, are defined from a pragma-dialectical perspective.[28] In dealing with unexpressed premises, for instance, first a differentiation is made between the 'logical minimum', i.e., the 'associated conditional' ('if premise, then conclusion'), and the 'pragmatic optimum', i.e., a further

[24] An integration of the Searlean speech act conditions and the Gricean conversational maxims in a set of 'rules of language use' is proposed in van Eemeren and Grootendorst [13, pp. 49–55], [15, ch. 4].

[25] See, e.g., Jackson and Jacobs [35] and Jacobs and Jackson [36, 37, 38].

[26] For a brief survey of the various approaches to the analysis of discourse and their empirical basis, see van Eemeren, Grootendorst, Jackson, and Jacobs [19, pp.50–59].

[27] Even a discourse that is clearly argumentative will in many respects not correspond to the ideal model of a critical discussion — or at least not directly and completely.

[28] At an introductory level these terms and concepts are explained in van Eemeren, Grootendorst and Snoeck Henkemans [18]. See also van Eemeren and Grootendorst [13] and van Eemeren [6].

specification or generalization of the associated conditional justified by the context and other relevant pragmatic considerations.[29] And in analyzing the argumentation structure, the multiple, coordinative and subordinative structures that are distinguished are associated with different kinds of responses to the critical questions the arguer anticipates, or responds to, when supporting a standpoint.[30] In turn, these critical questions are associated with the argument schemes that are used: they depend on whether the individual arguments and standpoints are connected by means of a causal, symptomatic or comparison relation.[31]

The elements included in an analytic overview are immediately relevant to the evaluation of argumentative discourse. If it is unclear what the difference of opinion is, there is no way of telling whether the difference has been resolved. If it is unclear which positions the parties have adopted, it will be impossible to tell in whose favor the discussion has ended. If implicit or indirect reasons and standpoints are not taken into account, crucial arguments may be overlooked and the evaluation is inadequate. If the structure of argumentation in favor of a standpoint is not exposed, it cannot be judged whether the arguments put forward in defense of the standpoint constitute a coherent and proper whole. If the argument schemes employed in supporting the various standpoints and sub-standpoints are not recognized, it cannot be determined whether the links between the individual arguments and the standpoints are equal to criticism.

9 Analytic Transformations in Reconstructing Argumentative Discourse

Generally, in argumentative discourse much remains implicit. Not only is there seldom any mention of the discussion rules or all the common starting points, but also other structural aspects of the resolution process are generally not indicated.[32] Because they are considered self-evident, but also for less honorable reasons, certain indispensable elements of the resolution process are often left unexpressed, including the exact nature of the disagreement, the division of roles, the relation between the arguments put forward in defense of a standpoint, the way in which the premises are supposed to support the standpoint, and even some of the premises. These elements usually are, sometimes in disguise, concealed in the discourse and need to be recovered in the analysis.

A reconstructive analysis of argumentative discourse as favored in pragma-dialectics entails a number of specific analytic operations that are instrumental in identifying the

[29] For the analysis of unexpressed premises, see van Eemeren and Grootendorst [13, pp. 60–72].

[30] For a discussion of the argumentation structures, see van Eemeren and Grootendorst [13, pp. 73–89].

[31] For a discussion of the argument schemes, see van Eemeren and Grootendorst [13, pp. 94–102].

[32] The implicit and unclear way in which the various stages of a critical discussion often appear in argumentative discourse, distorted and accompanied by diversions, should neither give rise to the premature conclusion that the discourse is deficient nor to the superficial conclusion that the ideal model of critical discussion is not realistic. The former is contradicted by pragmatic insight concerning ordinary discourse, the latter by dialectical insight concerning resolving differences of opinion. See van Eemeren and Grootendorst [10, ch. 4],[13, ch. 5]; and van Eemeren, Grootendorst, Jackson, and Jacobs [19, ch. 3].

elements in the discourse that play a part in resolving a difference of opinion. Each type of transformation represents a particular way of reconstructing part of the discourse in terms of a critical discussion.[33] The transformations are analytic tools for the externalization of participant commitments that are to be taken into account in an evaluation of the merits and demerits of the discourse. Due to the transformations, the discourse as it is written down or transcribed from a tape and the discourse that is reconstructed may differ in several respects. Depending on the transformations that are carried out, these differences can be characterized as resulting from deletion, addition, permutation, or substitution.

The transformation of deletion entails identifying elements in the discourse that are not relevant to resolving the difference of opinion, such as immaterial interruptions and sidelines, and omitting these elements in the analysis. Any dysfunctional repetitions that merely repeat the same message are also omitted. This transformation amounts to the removal of information that is redundant, superfluous, or otherwise irrelevant to the resolution goal.

The transformation of addition entails a process of completion. This transformation consists in supplementing the discourse as it is explicitly presented with those elements that are left implicit but are immediately relevant to the resolution of the dispute. Addition amounts to making elliptical elements and presuppositions explicit and supplementing moves that are not made explicitly in the text but are necessary for the discourse to make sense, such as the implicit arguments that are usually called unexpressed premises.

The transformation of permutation entails ordering and rearranging elements from the original discourse in such a way that the process of resolving the difference of opinion is set down as clearly as possible. In a pragma-dialectical analysis, the elements that are directly relevant to the resolution of the difference are recorded in the order that is most appropriate for the evaluation of the discourse. Unlike a descriptive record, the analysis need not necessarily follow the order of events in the discourse. Sometimes, the actual chronology can be retained; sometimes a rearrangement is called for to portray the resolution process. Overlap between different stages of a critical discussion is readjusted, just as anticipatory moves and references to earlier phases of the discourse. In this endeavor, confrontational elements that in the discourse are postponed until the conclusion are moved to the confrontation stage and argumentative moves that are advanced during the confrontation are put in their proper place in the argumentation stage.

The transformation of substitution involves an attempt to produce an explicit and clear presentation of the elements that are potentially instrumental in resolving the difference of opinion. Ambiguous or vague formulations are replaced by well defined and more precise standard phrases, giving elements that fulfill exactly the same function in the discourse but are phrased differently the same formulation. Different formulations of the same standpoint, for instance, are recorded in the same way and rhetorical questions that function as standpoints or arguments are noted as such. This process of translating elements from the discourse into standard phrases amounts to substituting pre-theoretical formulations of colloquial speech with formulations that are theoretically meaningful in the technical language of pragma-dialectics.

[33] See van Eemeren, Grootendorst, Jackson, and Jacobs [19, ch. 4].

In analytic practice, these reconstruction transformations are often carried out together in a cyclic process. For example, in reconstructing certain non-assertive speech acts as indirect standpoints, both the transformations of substitution and addition are carried out: a directive may thus first be reconstructed as an indirect assertive by means of the substitution transformation and then its communicative function of a standpoint is explicitly added by means of the addition transformation. Because it may become clear after a transformation has been carried out that some other transformation is also required and justified, the reconstruction process is recurrent and the analysis can be said to have a cyclic character.

For an illustration of the use of transformations in cases of indirectness, we take a closer look at the transformations by means of substitution and addition. In speech act theory, it is a recognized fact that in ordinary discourse the communicative function — or, as Searle calls it, 'illocutionary force' — of a speech act is not, as a rule, explicitly expressed. In many cases, this does not present much of a problem. The listener or reader is often directed to the desired interpretation by means of verbal indicators such as 'since' or 'therefore'. In the absence of such indicators the verbal and nonverbal context usually provide sufficient clues. Indirectness, however, can pose a problem. The following piece of discourse is an example:

> Let's take a cab. You don't want to be late for the show, do you?

In a resolution-oriented reconstruction the analyst would without any doubt say that this is argumentation, but where is the standpoint and what constitutes the argumentation? The standpoint is to be found in the first sentence, the second contains the argumentation. At first sight, however, the first speech act has the communicative function of a proposal, the second speech act that of a question. How can the attribution of the function of a standpoint to the first sentence, and that of argumentation to the second, be justified?

As speech act theory indicates, performing a proposal presupposes that the speaker believes it to be a good proposal. According to the correctness conditions for the performance of a proposal, the speaker wants the proposal to be accepted by the listener; otherwise it would be pointless. One way to get the proposal accepted would be to show that it is in the listener's interest. By asking rhetorically whether the listener wants to be late for the show, the speaker indirectly provides a possibly conclusive reason: The speaker knows very well that the listener does not want to be late (assuming the unexpressed premise that not taking a cab would cause this unwanted effect). By adding the rhetorical question to the proposal, the speaker tries to resolve a potential dispute in advance. This explains how his proposal can be transformed into the standpoint that it is wise to take a cab, and his rhetorical question into the argument that otherwise they will be late for the show (which is undesirable).[34] This reconstruction should suffice to show the merits of a pragmatic perspective in helping to get the transformations of substitution and addition carried out properly. Without speech act theory, no satisfactory analysis can be given.

[34] There is a difference between these two cases in the degree of 'conventionalization'. The rhetorical question is, as such, highly conventionalized, whereas the indirectness of the proposal is not. Only in a well-defined context indirectness can be easily detected and correctly interpreted. See van Eemeren and Grootendorst [13, pp. 56–59].

10 Rules for Critical Discussion

In pragma-dialectics, the critical norms of reasonableness authorizing the speech acts performed in the various stages of a critical discussion are accounted for in a set of dialectical rules. Taken together, the model and the rules constitute a theoretical definition of a critical discussion. In a critical discussion, the protagonists and the antagonists of the standpoints at issue not only go through all four stages of the resolution process, but they must also observe in every stage all the rules that are instrumental in resolving a difference of opinion.[35] The dialectical procedure proposed by van Eemeren and Grootendorst in *Speech Acts in Argumentative Discussions* [10] states the rules that are constitutive for a critical discussion in terms of the performance of speech acts.[36] They cover the entire argumentative discourse by stating all the norms that are pertinent to resolving a difference of opinion, ranging from the prohibition to prevent each other from expressing any position one wishes to assume in the confrontation stage, to the prohibition to generalize the result of the discussion in the concluding stage.

Proposing an ideal model with rules for critical discussion may lead to running the risk of being identified with striving for an unattainable utopia. The primary function of the pragma-dialectical model, however, is a different one. By systematically indicating what the rules for conducting a critical discussion are, the model provides those who want to fulfill the role of reasonable discussants with a series of guidelines. Though formulated on a higher level of abstraction and based on a clearly articulated philosophical ideal, they may be to a great extent identical to the norms the discussants would like to see observed anyway.

The pragma-dialectical rules for critical discussion that are to be followed in order to conduct the discussion effectively are to be judged for their capacity to serve this purpose well — their ‘problem-validity’. In order for the rules to be of practical significance, they must also be intersubjectively acceptable — so that they can acquire ‘conventional validity’.[37] The claim that these rules are acceptable is neither based on

[35] If the rules of the pragma-dialectical discussion procedure are regarded as first order conditions for having a critical discussion, the internal conditions for a reasonable discussion attitude can be viewed as ‘second order’ conditions relating to the state of mind the discussants are assumed to be in. In practice, people’s freedom to satisfy the second order conditions is sometimes limited by psychological factors beyond their control, such as emotional restraint and personal pressure. There are also external, ‘third order’ conditions that need to be fulfilled in order to be able to conduct a critical discussion properly. They relate to the social circumstances in which the discussion takes place and pertain, for instance, to the power or authority relations between the participants and the discussion situation. Together, the second and third order conditions for conducting a critical discussion in the ideal sense are higher order conditions for resolving differences of opinion. Only if these conditions are satisfied critical reasonableness can be fully realized in practice.

[36] An improved version of the pragma-dialectical rules for critical discussion is to be found in van Eemeren and Grootendorst [15, ch. 6].

[37] The notions ‘problem-validity’ and ‘conventional validity’, based on insight developed by Crawshay-Williams [5], are introduced by Barth and Krabbe [3]. In van Eemeren and Grootendorst [11, 12, 13] an account is given of the problem-validity of the pragma-dialectical rules; their inter-subjective validity was examined (and to a great extent confirmed) in a series of experimental tests.

metaphysical necessity nor derived from any external authority or sacrosanct origin, but rests on their effectiveness when applied in resolving a difference of opinion. Because the rules have been drawn up to promote the resolution of differences of opinion, assuming that they are correctly formulated, they should be acceptable to anyone who has that aim in view. Viewed philosophically, the rationale for accepting the rules can therefore be characterized as pragmatic.

What sort of people will be willing to provide conventional validity to the discussion rules? They will be people who accept doubt as an integral part of their way of life and use criticism toward themselves and others to solve problems by trial and error. They use argumentative discourse as a means to detect weaknesses in viewpoints regarding knowledge, values and objectives, and eliminate these weaknesses where possible.[38] It should be borne in mind that the primary aim of a critical discussion is not to maximize agreement but to test contested standpoints as critically as possible.[39]

The pragma-dialectical procedure for conducting a critical discussion is too technical for immediate use in ordinary practice. For practical purposes, based on the critical insight expressed in this procedure, a code of conduct has therefore been developed for people who want to resolve their differences of opinion by means of argumentation. This code of conduct consists of ten basic requirements for reasonable behavior, profanely referred to as the Ten Commandments. I restrict myself here to presenting the succinct recapitulation of the rules for critical discussion given in the Ten Commandments.

11 The Ten Commandments of Critical Discussion

The first commandment of the code of conduct is the freedom rule: Discussants may not prevent each other from advancing standpoints or from calling standpoints into question.

Commandment 1 is designed to ensure that standpoints, and doubt regarding standpoints, can be expressed freely. A difference of opinion cannot be resolved if it is not clear to the parties involved that there actually is a difference and what this difference involves. In argumentative discourse the parties must therefore have ample opportunity to make their positions known. In this way, they can make sure that the confrontation stage of a critical discussion is properly completed.

The second commandment is the obligation to defend rule: Discussants who advance a standpoint may not refuse to defend this standpoint when requested to do so. Commandment 2 is designed to ensure that standpoints that are put forward and called into question are defended against critical attacks. A critical discussion remains stuck in the opening stage and the difference of opinion cannot be resolved if the party who has advanced a standpoint is not prepared to fulfill the role of protagonist of this standpoint.

The third commandment is the standpoint rule: Attacks on standpoints may not bear on a standpoint that has not actually been put forward by the other party. Commandment 3 is primarily designed to ensure that attacks — and consequently defenses by means

[38] Such people, being opposed to protectionism of viewpoints and the immunization of any kind of standpoint against criticism, will reject all fundamentalist ‘justificationism’ (Letztbegründung). In taking this view, pragma-dialectics connects with formal dialectics as developed by Barth and Krabbe [3].

[39] See Popper [48, ch. 5, note 6].

of argumentation — relate to the standpoint that is indeed advanced by the protagonist. A difference of opinion cannot be resolved if the antagonist criticizes a different standpoint and the protagonist defends a different standpoint.

Commandment 4 is the relevance rule: Standpoints may not be defended by non-argumentation or argumentation that is not relevant to the standpoint. Commandment 4 is designed to ensure that the defense of standpoints takes place only by means of relevant argumentation. The difference of opinion that is at the heart of an argumentative discourse cannot be resolved if the protagonist advances arguments that do not pertain to the standpoint or resorts to rhetorical devices in which pathos or ethos take the place of logos.[40]

Commandment 5 is the unexpressed premise rule: Discussants may not falsely attribute unexpressed premises to the other party, nor disown responsibility for their own unexpressed premises. Commandment 5 ensures that the antagonist can examine every part of the protagonist's argumentation critically — also those parts that have remained implicit in the discourse. A difference of opinion cannot be resolved if the protagonist tries to evade the obligation to defend elements that he or she has left implicit, or if the antagonist misrepresents an unexpressed premise, for example, by exaggerating its scope.

Commandment 6 is the starting point rule: Discussants may not falsely present something as an accepted starting point or falsely deny that something is an accepted starting point. Commandment 6 is intended to ensure that when standpoints are attacked and defended, the starting points of the discussion are used in a proper way. Neither may something be presented as an accepted starting point if it is not, nor may it be denied that something is an accepted starting point if in fact it is. Otherwise it is impossible for a protagonist to defend a standpoint conclusively — and for an antagonist to attack that standpoint successfully — on the basis of commitments that can be viewed as concessions made by the other party.

Commandment 7 is the validity rule: Reasoning that in an argumentation is presented in an explicit and complete way may not be invalid in a logical sense. It is possible for antagonists and protagonists to determine whether the standpoints defended do indeed follow logically from the argumentation that is advanced only if the reasoning that is used in the argumentation is indeed verbalized in full. Commandment 7 is designed to ensure that protagonists who reason explicitly in resolving a difference of opinion use only reasoning that is valid in a logical sense.[41] When the reasoning is valid, the defended standpoint follows logically from the premises that are used, explicitly or implicitly, in the protagonist's argumentation. If not every part of the reasoning is fully expressed, commandment 7 does not apply.

Commandment 8 is the argument scheme rule: Standpoints may not be regarded conclusively defended if the defense does not take place by means of appropriate argument schemes that are applied correctly. Commandment 8 is designed to ensure that standpoints can indeed be conclusively defended if the protagonist and the antagonist agree

[40] This does not mean that advancing argumentation cannot be combined with, or even include, the use of pathos and ethos, or that relevant arguments cannot be suggested by, or implied in, apparently irrelevant arguments. For an overview of (the history of) classical rhetoric, and an explanation of the role of logos, ethos and pathos, see Kennedy [41].

[41] What is meant by 'valid in a logical sense' depends on the logical theory that is used.

on a method to test the soundness of the types of arguments that are used and are not part of the common starting point.[42] This implies that they must examine whether the argument schemes that are used are admissible in the light of what has been agreed upon in the opening stage, and whether they have been correctly fleshed out in the argumentation stage.

Commandment 9, bearing on the concluding stage, is the concluding rule: Inconclusive defenses of standpoints may not lead to maintaining these standpoints and conclusive defenses of standpoints may not lead to maintaining expressions of doubt concerning these standpoints. Commandment 9 is designed to ensure that in the concluding stage the protagonists and the antagonists correctly ascertain the result of the discussion. A difference of opinion is resolved only if the parties are in agreement that the defense of the standpoints at issue has been successful or has not been successful.

The tenth and last commandment is the general language use rule: Discussants may not use any formulations that are insufficiently clear or confusingly ambiguous, and they may not deliberately misinterpret the other party's formulations. Problems of formulation and interpretation can occur in any stage of a critical discussion. Commandment 10 is designed to ensure that misunderstandings arising from unclear, vague or equivocal formulations are avoided. A difference of opinion can only be resolved if each party makes a real effort to express its intentions as accurately as possible in a way that minimizes the chances of misunderstanding. Equally, a difference of opinion can only be resolved if each party makes a real effort not to misinterpret any of the other party's speech acts. Problems of formulation or interpretation may otherwise lead to a pseudo-difference or to a pseudo-solution.

12 Fallacies as Counterproductive Moves in Resolving Disagreement

A pragma-dialectical evaluation of argumentative discourse is aimed at determining the extent to which the various speech acts performed in the discourse are instrumental in resolving a difference of opinion. In order to achieve this goal, the evaluation needs to make clear which discussion moves hinder or obstruct a critical discussion. When an analytic overview has been compiled on the basis of a justified reconstructive analysis, a suitable point of departure has been created for such an evaluation of the discourse.

In principle, each of the pragma-dialectical discussion rules constitutes a distinct standard or norm for critical discussion. Any move constituting an infringement of any of the rules, whichever party performs it and at whatever stage in the discussion, is a possible threat to the resolution of a difference of opinion and must therefore (in this particular sense) be regarded as fallacious.[43] The use of the term 'fallacy' is then systematically connected with the rules for critical discussion and a fallacy is defined as a discussion move that violates in some specific way a rule for critical discussion applying to a particular discussion stage.

[42] See van Eemeren and Grootendorst [13, 94–102].

[43] The pragma-dialectical identification of fallacies is always conditional. An argumentative move may be regarded as a fallacy only if the discourse is correctly viewed as aimed at resolving a difference of opinion.

This approach to the fallacies, fleshed out by van Eemeren and Grootendorst [13] in *Argumentation, Communication, and Fallacies*, offers an alternative to the Standard Treatment of the fallacies that was criticized devastatingly by Hamblin [29].[44] Rather than considering the fallacies as belonging to an unstructured list of nominal categories that happen to have been inherited from the past or considering all fallacies as violations of one and the same (validity) norm, the pragma-dialectical approach differentiates a functional variety of norms. Depending on the rule that has been violated, a series of other norms than logical validity are taken into account. In this way, many of the traditional fallacies can be characterized more clearly and consistently, while 'new' fallacies are identified that went earlier unnoticed.

13 Violations of the Code of Conduct for Critical Discussion

When it comes to the detection of fallacies, a pragma-dialectical analysis proceeds in a number of steps. An utterance must first be interpreted as a particular kind of speech act performed in a context of discourse aimed at resolving a difference of opinion. Then it must be determined whether the performance of this speech act agrees with the rules for critical discussion. If the speech act proves to be a violation of any of the norms pertaining to a particular stage of the resolution process, the kind of violation must be typified by determining which specific criterion for satisfying the norm has not been met.

The freedom rule (1) can be violated — in the confrontation stage — in various ways, both by the protagonist and the antagonist. A party can impose certain restrictions on the standpoints that may be advanced or called into question; a party can deny an opponent the right to advance a certain standpoint or to criticize a certain standpoint. Violations of the first kind mean that particular standpoints are declared sacrosanct or that some standpoints are in fact excluded from discussion. Violations of the first kind are directed at the opponent personally and aim at eliminating the opponent as a serious discussion partner. This may be done by putting pressure on the opponent, threatening that person with sanctions (argumentum ad baculum), or by playing on the opponent's feelings of compassion (argumentum ad misericordiam), but also by discrediting the opponent's expertise, impartiality, integrity or credibility (argumentum ad hominem).

The obligation to defend rule (2) can be violated — in the opening stage — by the protagonist by evading or shifting the burden of proof. In the first case, the protagonist attempts to create the impression that there is no point in calling the standpoint into question, and no need to defend it, by presenting the standpoint as self-evident by giving a personal guarantee of the correctness of the standpoint (variant of argumentum ad verecundiam) or by immunizing the standpoint against criticism. In the last case, the protagonist challenges the opponent to show that the protagonist's standpoint is wrong (variant of argumentum ad ignorantiam) or that the opposite standpoint is right.

The standpoint rule (3) can be violated — in all stages — by the protagonist or the antagonist. In a discussion about a mixed difference of opinion they can do so by imputing a fictitious standpoint to the other party or distorting the other party's standpoint (straw man). The first effect can be achieved by emphatically advancing the opposite

[44] For an overview of the pre-Hamblin and post-Hamblin theoretical approaches to the fallacies, see [7].

as one's own standpoint or by creating an imaginary opponent; the second by taking utterances out of context by oversimplification (ignoring nuances or qualifications) or by exaggeration (making something absolute or generalizing).

The relevance rule (4) can be violated — in the argumentation stage — by the protagonist in two ways: by putting forward argumentation that does not refer to the standpoint advanced in the confrontation stage (irrelevant argumentation or ignoratio elenchi); second, by defending a standpoint using non-argumentative means of persuasion. Playing on the emotions of the audience (variant of argumentum ad populum) and parading one's own qualities (variant of argumentum ad verecundiam) are examples. If the audience's positive or negative emotions (such as prejudice) are exploited, pathos replaces logos. For this reason, such violations of the relevance rule are sometimes called pathetic fallacies. If protagonists wrongly attempt to get their standpoints accepted by the opponent because of their authority in the eyes of the audience due to their expertise, credibility, integrity, or other qualities, ethos replaces logos; for this reason, such violations of the relevance rule are sometimes called ethical fallacies.

The protagonist can violate the unexpressed premise rule (5) — in the argumentation stage — by denying an unexpressed premise, and the antagonist can violate the same rule by distorting an unexpressed premise. In denying an unexpressed premise ("I never said that"), the protagonist in effect tries to evade the responsibility assumed in argumentation by denying a commitment to an unexpressed premise that is correctly reconstructed as such. Antagonists are guilty of the fallacy of distorting an unexpressed premise if they have produced a reconstruction of a protagonist's unexpressed premise that goes beyond the 'pragmatic optimum' to which the protagonist can actually be held, given the verbal and nonverbal context.

The starting point rule (6) can be violated — in the argumentation stage — by the protagonist's falsely presenting something as a common starting point or by the antagonist's denying a premise representing a common starting point. By falsely presenting something as a common starting point, the protagonist tries to evade the burden of proof; the techniques used for this purpose include falsely presenting a premise as self-evident, enveloping a proposition in a presupposition of a question (many questions), concealing a premise in an unexpressed premise, and advancing argumentation that amounts to the same thing as the standpoint (petitio principii, also called begging the question or circular reasoning). By denying a premise representing a common starting point, the antagonist denies the protagonist the opportunity to defend his or her standpoint ex concessis, which is a denial of a conditio sine qua non for all successful argumentation.

The validity rule (7) can be violated — in the argumentation stage — by the protagonist in a variety of ways. Some cases of logical invalidity occur regularly and are often not immediately recognized. Among them are confusing a necessary condition with a sufficient condition (or vice versa) in arguments with an 'If..., then...'-premise (affirming the consequent, denying the antecedent). Other violations amount to erroneously attributing a (relative or structure-dependent) property of a whole to its constituent parts or vice versa (fallacies of division and composition).

The argument scheme rule (8) can be violated — in the argumentation stage — by the protagonist by relying on an inappropriate argument scheme or using an appropriate argument scheme incorrectly. The violations can be classified according to the three main categories of argument schemes: symptomatic argumentation of the ‘token’ type, where there is a relation of concomitance between the premises and the standpoint (“Daniel is an actor [and actors are typically vain], so he is certainly vain”), comparison argumentation of the ‘similarity’ type, where the relation is one of resemblance (“The measure I would like to take is fair, because the case we had last year was also dealt with in this way [and the one case is similar to the other]”, and instrumental argumentation of the ‘consequence’ type, where the relation is one of causality (“Because Tom has been drinking an excessive amount of whiskey [and drinking too much alcohol leads to a terrible headache], he must have a terrible headache”).

Symptomatic argumentation is used incorrectly if, for instance, a standpoint is presented as right because an irrelevant or quasi-authority says so (special variant of argumentum ad verecundiam) or because everybody thinks it is right (populist variant of argumentum ad populum and also a special variant of argumentum ad verecundiam), or if a standpoint is a generalization based upon observations that are not representative or insufficient (hasty generalization or secundum quid). Comparison argumentation is used incorrectly, if, for instance, in making an analogy the conditions for a correct comparison are not fulfilled (false analogy). Finally, instrumental argumentation is used incorrectly if, for instance, a descriptive standpoint is being rejected because of its undesired consequences (argumentum ad consequentiam); a cause-effect relation is inferred from the mere observation that two events take place one after the other (post hoc ergo propter hoc); or it is unjustifiably suggested that by taking a proposed course of action one will be going from bad to worse (slippery slope).

The concluding rule (9) can be violated — in the concluding stage — by the protagonist concluding that a standpoint is true merely because it has been successfully defended (making an absolute of the success of the defense) or by the antagonist concluding from the fact that it has not been proved that something is the case, that it is not the case, or from the fact that something has not been proved not to be the case, that it is the case (making an absolute of the failure of the defense or special variant of argumentum ad ignorantiam). In making an absolute of the success of the defense, the protagonist commits a double error: first, the unjustified status of established fact, the truth of which is beyond discussion, is ascribed to the common starting points; secondly, in doing so, a successful defense is erroneously invested with an objective rather than inter-subjective status. In making an absolute of the failure of the defense, the antagonist commits a double error: first, the roles of antagonist and protagonist are confused; second, it is mistakenly assumed that a discussion must always end in a victory for either a positive or a negative standpoint, so that not having the positive standpoint automatically means adopting the negative standpoint, and vice versa, ignoring the possibility of entertaining a ‘zero’ standpoint.[45]

The language use rule (10) can be violated — in all stages — by the protagonist or the antagonist by taking undue advantage of unclearness (fallacy of unclearness) or ambiguity (fallacy of ambiguity, equivocation, amphiboly). Various sorts of unclearness can occur: unclearness resulting from the structuring of the text, from implicitness, indefiniteness, unfamiliarity, vagueness, and so on. Again, there are various sorts of ambiguity:

[45] A ‘zero’ standpoint occurs in a non-mixed difference of opinion when the other party only has doubts about the acceptability of the standpoint. See van Eemeren and Grootendorst [13, pp. 13–25].

referential ambiguity, syntactic ambiguity, semantic ambiguity, and so on. The fallacy of ambiguity is closely related to the fallacy of unclearness; it can occur on its own but also in combination with other fallacies (such as the fallacies of composition and division).

This brief overview may suffice to show that the pragma-dialectical analysis of the traditional fallacies as violations of the rules for critical discussion is more systematic than the Standard Treatment criticized by Hamblin. Instead of being given ad hoc explanations, all the fallacies are understood as falling under one or more of the rules for critical discussion. Fallacies that only were lumped nominally together in the traditional categories are either shown to have something in common or they are clearly distinguished. Genuinely related fallacies that were before separated are brought together. Distinguishing two variants of the argumentum ad populum — one a violation of relevance rule 4, the other of argument scheme rule 8 — makes clear, for instance, that these variants are in fact not of the same kind. Analyzing one particular variant of the argumentum ad verecundiam and one particular variant of the argumentum ad populum as violations of the argument scheme rule make clear that these variants are of the same kind when viewed from the perspective of resolving a difference of opinion.

The analytic overview also reveals that the pragma-dialectical approach makes it possible to identify so far non-recognized and unnamed 'new' obstacles to resolving a difference of opinion: declaring a standpoint sacrosanct (violation of freedom rule 1), evading the burden of proof by immunizing a standpoint against criticism (violation of obligation to defend rule 2) or falsely presenting a premise as self-evident (violation of starting point rule 6), denying an unexpressed premise (violation of unexpressed premise rule 5), denying an accepted starting point (violation of starting point rule 6), falsely presenting something as a common starting point (violation of starting point rule 6), making an absolute of the success of the defense (violation of concluding rule 9), and so on.

14 Making Use of Insight in Strategic Maneuvering

However justified it may be to view pragmatics as the modern version of rhetoric, certain attainments of classical rhetoric are then neglected that are vital to the study of argumentation. According to van Eemeren and Houtlosser, the pragma-dialectical method of analyzing and evaluating argumentative discourse can be enriched by a systematic integration of rhetorical insight in the dialectical theoretical framework [20, 21, 22, 23, 24]. To remedy the existing separation between dialectic and rhetoric, it is necessary to realize that the two views are not incompatible, but can even be complementary.[46] Gener-

[46] Regrettably, in academic practice there is still a yawning conceptual gap and lack of understanding between the protagonists of a dialectical approach and a rhetorical approach. As generally perceived, in Greek Antiquity the difference amounted initially to a division of labor. According to Toulmin [55], after the 17th century's Scientific Revolution, the division became 'ideological' and resulted in two mutually isolated paradigms, which were regarded incompatible. Rhetoric has become a field of study in the humanities for scholars interested in communication, discourse analysis and literature. Dialectic was first incorporated in the exact sciences and disappeared with the further formalization of logic in the nineteenth century for a long time almost altogether from sight. Until recently, rhetoricians largely ignored the results of dialectical theorizing, and the other way around. The papers in van Eemeren and Houtlosser [25] are part of an effort to stimulate a rapprochement.

ally, in argumentative discourse it is not the arguers' sole aim to conduct the discussion in a reasonable way, but also to win the discussion by having their point accepted. The arguers' rhetorical attempts to have things their way are incorporated in their efforts to realize their dialectical aspiration of resolving the difference of opinion in accordance with the standards pertaining to a critical discussion.

Viewed pragma-dialectically, in argumentative discourse the parties are in every stage of the resolution process out for the optimal rhetorical result at the stage they are going through, but may at the same time be presumed to hold to the dialectical objective of that discussion stage. Thus the dialectical aim of each of the four stages of the resolution process may be taken to have its rhetorical analogue. To reconcile the simultaneous pursuit of these two different aims, the arguers make use of strategic maneuvering aimed at diminishing the potential tension between the two [21]. The basic aspects of strategic maneuvering distinguished in pragma-dialectics are: (1) making an expedient selection from the 'topical potential', i.e., the set of available alternatives in a certain discussion stage; (2) adapting one's contribution optimally to 'audience demand', i.e., the specific preferences and expectations of the listener(s) or reader(s); and (3) using the most effective 'presentational devices', i.e., the various stylistic and other verbal and non-verbal means of conveying a message. If the selection results in a concerted succession of moves, in which the choices regarding the three aspects are coordinated, a full-fledged argumentative strategy is used.[47]

A pragma-dialectical analysis can benefit in several respects from using this conception of strategic maneuvering in reconstructing argumentative discourse. Taking the strategic maneuvering into account provides a clearer view of the rhetorical dimension of the discourse, so that a more comprehensive grasp is gained of argumentative reality. Through the more thorough and subtle understanding of the rationale behind the various moves made in the discourse the analysis becomes more profound. And by combining such rhetorical insight with the pragma-dialectical insight already achieved in the reconstruction process, the analysis can be better justified.[48]

15 Fallacies as Derailments of Strategic Maneuvering

The strategic maneuvering that takes place in argumentative discourse to maintain the balance between dialectical and rhetorical objectives may sometimes lead to inconsistencies and 'derail'. Such derailments generally coincide with the non-constructive moves in argumentative discourse that are traditionally known as fallacies. One of the crucial problems in detecting fallacies is how sound and fallacious argumentative discourse can be distinguished. In pragma-dialectics, argumentative moves are considered sound if they are in agreement with the rules applying to the stage of a critical dis-

[47] What the best way of strategic maneuvering is depends in the last instance always on the contextual limits set by the dialectical situation, the audience that is to be persuaded, and the usable linguistic repertoire.

[48] The pragma-dialectical theory as originally developed by van Eemeren and Grootendorst [10, 13, 15] can be seen as a dialectical approach to argumentation that keeps an open eye for rhetorical aspects of argumentative reality by studying argumentative discourse from a pragmatic perspective, but does not explicitly take insight from rhetoric into account.

cussion in which they are made and fallacious if they violate any of these rules.[49] To be able, however, to determine systematically for all stages of the resolution process whether or not certain argumentative moves violate a rule, clear criteria are required for deciding when exactly a certain norm encapsulated in a particular discussion rule has been violated. The concept of strategic maneuvering can be of help in identifying such criteria.

In principle, all the moves made in argumentative discourse are motivated both by the aim of arguing reasonably and the aim of having things one's own way, but these aspirations are not always in perfect balance. On the one hand, speakers or writers may neglect their persuasive interests, e.g., for fear of being perceived as unreasonable; on the other hand, they may neglect their commitment to the critical ideal due to their assiduity to win the other party over to their side. Neglect of persuasiveness will harm the arguer but not the adversary, and is therefore not 'condemnable' as being fallacious. However, if a party allows its commitment to a reasonable exchange of argumentative moves to become overruled by the aim of persuading the other party, the strategic maneuvering derails because the other party becomes the victim and the maneuvering is then condemnable for being fallacious.[50]

Each mode of strategic maneuvering is associated with a certain continuum of sound and fallacious acting and often the demarcation line between the two can only be determined contextually.[51] The criteria for determining fallacious strategic maneuvering can be more fully and systematically determined if we are able to rely on a well-motivated classification of the diverse modes of strategic maneuvering in the various discussion stages. If, for the confrontation stage, for instance, it can be established in which ways the parties may shape the issues on which they differ and the positions they assume to their own advantage, and the modes of strategic maneuvering can be specified that serve certain 'local' and stage-related rhetorical aims, it becomes possible to investigate more precisely which soundness conditions apply. By relating the modes of strategic maneuvering concerned to the dialectical aim of the confrontation stage, appropriate criteria can be established that need to be taken into account in deciding whether or not a particular instance of strategic maneuvering has got derailed and a fallacy has been committed.

[49] This approach differs from approaches to the fallacies, such as Biro and Siegel's [4] and Johnson's [39], that give precedence to — absolute — epistemological considerations, and Willard's [60] and Leff's [42], that rely on empirical — and relativistic — social considerations.

[50] Because a party who commits a fallacy will at the same time uphold a commitment to complying with the rules of critical discussion, an assumption of reasonableness is conferred on every discussion move (see also Jackson, [34]). This assumption is operative even when a particular way of maneuvering violates a certain discussion rule. This explains why fallacies are often not immediately manifest or apparent to others. Echoing the definition of a fallacy criticized by Hamblin [29, p. 12], one can say that the maneuvering then still 'seems' to obey the rules of critical discussion, although in fact it does not. The approach of fallacies as derailments of strategic maneuvering can thus be of help in explaining the deceptive character of (some of) the fallacies.

[51] There are some specific derailments of strategic maneuvering that can be generally pinned down as clear-cut violations of a certain rule applying to a particular discussion stage, but they are exceptional.

To illustrate how the identification of criteria for demarcating fallacious and sound modes of strategic maneuvering may proceed, we take an example from an ‘advertorial’ in which Shell defends its not pulling out of Nigeria’s Liquefied Natural Gas project:

> If we do so now, the project will collapse. [...] A cancellation would certainly hurt the thousands of Nigerians who will be working on the project, and the tens of thousands more benefiting in the local economy. The environment, too, would suffer, with the plant expected to cut greatly the need for gas flaring in the oil industry.

Shell chooses its arguments for not pulling out of the project straight from its opponents’ political concerns for the people of Nigeria and the environment, so that its strategic maneuvering is characterized by the use of conciliatio, i.e., convincing the other party by exploiting its own views. In view of its opponents’ professed concerns, at the proposition level Shell can be sure of acceptance. But how does the oil company proceed to ensure the opponents’ acceptance of the justificatory potential of the two arguments for a standpoint that is precisely the opposite of their own? The company lends support to the view that the arguments of its opponents have an overriding justificatory potential for its own standpoint by claiming that there is a causal relation between Shell’s pulling out of the project and a deterioration of the human and environmental circumstances. In spite of the use of the word ‘certainly’, Shell does not really deter the reader from questioning the supposed causal link, so that it cannot be maintained that a derailment of strategic maneuvering has actually taken place, and there is no sufficient reason to accuse Shell of question begging. The use of conciliatio is a derailment of strategic maneuvering only if it is simply assumed that an argument that has been taken over has an unquestioning justificatory potential for the standpoint at issue and there is no room left for criticizing this presupposition.

16 Conclusion

In the pragma-dialectical approach, argumentation is studied from a communicative perspective by means of a comprehensive research program that has a descriptive as well as a normative dimension. The methodological principles of functionalization, socialization, externalization and dialectification are realized in the ideal model of a critical discussion that portrays the distribution of speech acts over the various stages of the process of resolving a difference of opinion. The rules for critical discussion pertaining to these speech acts constitute distinct standards for argumentative conduct which can be summarized as a code of conduct for critical discussion. Any infringement of any of the rules is a possible threat to the resolution of a difference of opinion and must therefore be regarded as an incorrect discussion move, which can be analyzed as a fallacy. The problem validity of the rules is judged, pragmatically, by their theoretical contribution to the resolution of a difference of opinion. In order to be effective, however, the rules must also be intersubjectively acceptable to those people who wish to resolve their differences by means of argumentative discourse. The intersubjective validity of the rules has been tested empirically by experiments aimed at determining systematically the extent to which they agree with the norms favored by ordinary language users when evaluating argumentative discourse. The pragma-dialectical method for analyzing argumentative discourse involves a systematic reconstruction of the discourse that results in an analytic overview containing all aspects of the discourse that are pertinent to the resolution of a difference of opinion. A recent crucial step in the development of this method was the introduction of the notion of strategic maneuvering, which refers to the perennial balancing between pursuing at the same time a resolution-minded dialectical objective and the rhetorical objective of having one's own position accepted. In the future, examining strategic maneuvering will no doubt lead to more refined and more thoroughly justified analyses. It will also lead to a better evaluation of derailments of strategic maneuvering. The criteria needed for identifying and evaluating potentially fallacious maneuvering must be determined in relation with the specific context in which the maneuvering takes place.

References

1. H. Albert. *Traktat über kritische Vernunft*. Mohr, 3rd edition, 1975. (Ist Edition 1967).
2. J. L. Austin. *How to Do Things with Words*. Clarendon Press, Oxford, 1962.
3. E. M. Barth and E. C. W. Krabbe. *From Axiom to Dialogue: A Philosophical Study of Logics and Argumentation*. Walter de Gruyter, Berlin/New York, 1982.
4. J. Biro and H. Siegel. Normativity, argumentation and an epistemic theory of fallacies. In F.H. van Eemeren, R. Grootendorst, J.A. Blair, and C.A. Willard, editors, *Argumentation Illuminated*, pages 85–103. Sic Sat, Amsterdam, 1992.
5. R. Crawshay-Williams. *Methods and Criteria of Reasoning: An Inquiry into the Structure of Controversy*. Routledge & Kegan Paul, London, 1957.
6. F. H. van Eemeren, editor. *Crucial Concepts in Argumentation Theory*. Amsterdam University Press, Amsterdam, 2001.
7. F. H. van Eemeren. Fallacies. In *Crucial Concepts in Argumentation Theory* [6].
8. F. H. van Eemeren, editor. *Advances in Pragma-Dialectics*. Sic Sat/Vale Press, Amsterdam/Newport News, VA., 2002.
9. F. H. van Eemeren, B. J. Garssen, and B. Meuffels. The unreasonableness of the ad baculum fallacy. In Th. Goodnight, editor, *Arguing Communication & Culture. Selected Papers from the Twelfth NCA/AFA Conference on Argumentation*, pages 343–350. National Communication Association, Washington, DC, 2002.
10. F. H. van Eemeren and R. Grootendorst. *Speech Acts in Argumentative Discussions: A Theoretical Model for the Analysis of Discussions Directed towards Solving Conflicts of Opinion*. Foris/Mouton de Gruyter, Dordrecht/Berlin, 1984.
11. F. H. van Eemeren and R. Grootendorst. Rationale for a pragma-dialectical perspective. *Argumentation*, 2:271–291, 1988. Also published in [14, pp. 11–28].
12. F. H. van Eemeren and R. Grootendorst. Rules for argumentation in dialogues. *Argumentation*, 2:499–510, 1988.
13. F. H. van Eemeren and R. Grootendorst. *Argumentation, Communication, and Fallacies: A Pragma-Dialectical Perspective*. Lawrence Erlbaum, Hillsdale, NJ, 1992.
14. F. H. van Eemeren and R. Grootendorst, editors. *Studies in Pragma-Dialectics*. Sic Sat, Amsterdam, 1994.
15. F. H. van Eemeren and R. Grootendorst. *A Systematic Theory of Argumentation. The Pragma-Dialectical Approach*. Cambridge University Press, Cambridge, 2004.
16. F. H. van Eemeren and B. Meuffels. Ordinary arguers' judgements on ad hominem fallacies. 2002. in van Eemeren [8], pages 45–64.
17. F. H. van Eemeren, B. Meuffels, and M. Verburg. The (un)reasonableness of the argumentum ad hominem. *Language and Social Psychology*, 19:416–435, 2000.

18. F.H. van Eemeren, R. Grootendorst, and A. F. Snoeck Henkemans. *Argumentation: Analysis, Evaluation, Presentation*. Lawrence Erlbaum, Mahwah, NJ, 2002.
19. F.H. van Eemeren, R. Grootendorst, S. Jackson, and S. Jacobs. *Reconstructing Argumentative Discourse*. The University of Alabama Press, Tuscaloosa/London, 1993.
20. F.H. van Eemeren and P. Houtlosser. Rhetorical rationales for dialectical moves: Justifying pragma-dialectical reconstructions. In J.F. Klumpp, editor, *Argument in a Time of Change: Definitions, Frameworks, and Critiques*, pages 51–56. National Communication Association, Annandale, VA. Proceedings of the Tenth NCA/AFA Conference on Argumentation. Alta, Utah, August 1997.
21. F.H. van Eemeren and P. Houtlosser. Strategic maneuvering: Maintaining a delicate balance. in [25], pages 131–159.
22. F.H. van Eemeren and P. Houtlosser. Rhetorical analysis within a pragma-dialectical framework. *Discourse Studies*, 1:479–497, 1999.
23. F.H. van Eemeren and P. Houtlosser. Managing disagreement: Rhetorical analysis within a dialectical framework. *Argumentation and Advocay*, 37:150–157, 2000.
24. F.H. van Eemeren and P. Houtlosser. Rhetorical analysis within a pragma-dialectical framework. *Argumentation*, 14:293–305, 2000.
25. F.H. van Eemeren and P. Houtlosser, editors. *Dialectic and Rhetoric: The Warp and Woof of Argumentation Analysis*. Kluwer Academic Publishers, Dordrecht, 2002.
26. F.H. van Eemeren, R. R. Grootendorst, and B. Meuffels. The skill of identifying argumentation. *Journal of the American Forensic Association*, 25:239–245, 1989. Also included in [14], pages. 119–129.
27. H. P. Grice. *Studies in the Way of Words*. Harvard University Press, Cambridge, MA, 1989.
28. J. Habermas. Vorbereitende bemerkungen zu einer theorie der kommunikativen kompetenz. In J. Habermas and H. Luhmann, editors, *Theorie der Gesellschaft oder Sozialtechnologie. Was leistet die Systemforschung?*, pages 107–141. Suhrkamp, Frankfurt, 1971.
29. C. L. Hamblin. *Fallacies*. Methuen, London, 1970. (Reprinted at Newport News: Vale Press).
30. A. F. Snoeck Henkemans. *Analysing Complex Argumentation. The Reconstruction of Multiple and Coordinatively Compound Argumentation in a Critical Discussion*. Sic Sat., Amsterdam, 1992.
31. J. Heritage. A change-of-state token and aspects of its sequential placement. In J.M. Atkinson and J. Heritage, editors, *Structures of Social Action*, Studies in Conversation Analysis, pages 299–346. Cambridge University Press, Cambridge, 1984.
32. P. Houtlosser. The speech act "advancing a standpoint". In Eemeren and Grootendorst [14], pages 165–171.
33. S. Jackson. "virtual standpoints" and the pragmatics of conversational argument. In *Argumentation Illuminated*, pages 260–269. Sic Sat, 1, Amsterdam, 1992.
34. S. Jackson. Fallacies and heuristics. In *Analysis and Evaluation. Proceedings of the Third ISSA Conference on Argumentation*, volume II, pages 260–269. Sic Sat., Amsterdam, 1995. University of Amsterdam, June 21–24, 1994.
35. S. Jackson and S. Jacobs. Of conversational argument: pragmatic bases for the enthymeme. *Quarterly Journal of Speech*, 66:251–265, 1980.
36. S. Jacobs and S Jackson. Argument as a natural category: the routine grounds for arguing in natural conversation. *Western Journal of Speech Communication*, 45:118–132, 1981.
37. S. Jacobs and S Jackson. Conversational argument: a discourse analytic approach. In J. R. Cox and C. A. Willard, editors, *Advances in Argumentation Theory and Research*, pages 205–237. Southern Illinois University Press, Carbondale, IL, 1982.
38. S. Jacobs and S Jackson. Strategy and structure in conversational influence attempts. *Communication Monographs*, 50:285–304, 1983.

39. R. Johnson. *Manifest Rationality. A Pragmatic Theory of Argument*. Lawrence Erlbaum, Mahwah, NJ., 2000.
40. P. N. Johnson-Laird. *Mental Models. Towards a Cognitive Science of Language, Inference, and Consciousness*. Canbridge University Press, Cambridge, 1983.
41. G. A. Kennedy. *A New History of Classical Rhetoric*. Princeton University Press, Princeton, NJ, 1994.
42. M. Leff. Rhetoric and dialectic in the twenty-first century. *Argumentation*, 14:241–254, 2000.
43. S. C. Levinson. *Pragmatics*. Cambridge University Press, Cambridge, 1983.
44. P. Lorenzen and K. Lorenz. *Dialogische Logik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1978.
45. A. Naess. *Communication and Argument. Elements of Applied Semantics*. Allen & Unwin, London, 1966. (English translation of *Om meningsytring. En del elementaere logiske emner*, Oslo: Universitetsforlaget, 1947).
46. R. Nisbett and L. Ross. *Human Inference: Strategies and Shortcomings of Social Judgement*. Prentice-Hall, Englewood Cliffs, NJ, 1980.
47. D. J. O'Keefe. The persuasive effects of variation in standpoint articulation. In Eemeren [8], pages 65–82.
48. K. R. Popper. *The Open Society and its Enemies*. Princeton University Press, Princeton, NJ, 5th edition, 1971.
49. K. R. Popper. *Objective Knowledge. An Evolutionary Approach*. Clarendon Press, Oxford, 1972.
50. K. R. Popper. *Conjectures and Refutations. The Growth of Scientific Knowledge*. Routledge & Kegan Paul, London, 1974.
51. J. R. Searle. *Speech Acts. An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.
52. J. R. Searle. *Expression and Meaning. Studies in the Theory of Speech Acts*. Cambridge University Press, Cambridge, 1979.
53. S. E. Toulmin. *The uses of argument*. Cambridge University Press, Cambridge, 1958.
54. S. E. Toulmin. *Knowing and acting. An invitation to philosophy*. Macmillan, 1976.
55. S. E. Toulmin. *Return to Reason*. Harvard University Press, 2001.
56. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, 1995.
57. J. W. Wenzel. Jürgen habermas and the dialectical perspective on argumentation. *Journal of the American Forensic Association*, 16:83–94, 1979.
58. C. A. Willard. *Argumentation and the Social Grounds of Knowledge*. The University of Alabama Press, Tuscaloosa, 1983.
59. C. A. Willard. *A Theory of Argumentation*. The University of Alabama Press, Tuscaloosa, 1989.
60. C. A. Willard. *Liberal Alarms and Rhetorical Excursions. A New Rhetoric for Modern Democracy*. University of Chicago Press, Chicago, 1995.

A Logic of Abstract Argumentation

Guido Boella[1], Joris Hulstijn[2], and Leendert van der Torre[3]

[1] Universitá di Torino
[2] Vrije Universiteit, Amsterdam
[3] CWI Amsterdam and Delft University of Technology

Abstract. In this paper we introduce a logic of abstract argumentation capturing Dung's theory of abstract argumentation, based on connectives for attack and defend. We extend it to a modal logic of abstract argumentation to generalize Dung's theory and define variants of it. Moreover, we use the logic to relate Dung's theory of abstract argumentation to more traditional conditional and comparative formalisms, and we illustrate how to reason about arguments in meta-argumentation.

1 Introduction

Dung's theory of abstract argumentation [7] is popular in agent theory. For example, Prakken and Vreeswijk note that on the one hand it unifies theories on argumentation [14], and on the other hand it unifies theories of non-monotonic reasoning [6]. However, it has also been criticized. For example, Horty observes that the pattern called reinstatement is an integrated part of Dung's theory, whereas this pattern has been criticized in non-monotonic reasoning [9]. In multiagent systems argumentation theory is used for dialogue, for example by Parsons *et al* [12], because there is no commonly known truth to refer to. In other words, argumentation is all there is to establish agreement. This is analogous to the situation in legal reasoning.

However, argumentation theory is hardly used in agent technology. For example, it is not used for model checking agent dialogues [17]. There are two related problems. First, various researchers have claimed that the model-theoretic approach is not suitable for argumentation, such that the model-theoretic approach to argumentation has not been developed. Secondly, the relation between argumentation theory and other formal systems has not been studied, such that that existing technologies cannot be used for argumentation. There is a tendency within argumentation theory to develop specialized procedures rather than to connect to existing technologies.

In this paper we study argumentation theory from a logical point of view, using model-theoretic semantics, and we relate it to other formal systems. From a formal point of view, Dung does not consider conditionals used in traditional argumentation and non-monotonic reasoning, such as for example $a \rightarrow b$: a is an argument for (supports) b. Instead, the central concept studied in abstract argumentation is a binary *attack* relation among arguments. In this paper we represent it by '$\rhd$'. We write $a \rhd b$ for argument a attacks argument b.

Though both $a \rightarrow b$ and $a \rhd b$ are binary connectives, they have distinct logics. For example, whereas most conditional logics accept the identity rule, $a \rightarrow a$, we definitely do not have that all arguments attack itself: $a \not\rhd a$. Moreover, whereas a conditional connective '$\rightarrow$' might satisfy the transitivity or the cut rule, this does not make sense for the

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 29–41, 2006.

attack connective '$\rhd$'. The latter also distinguishes the attack connective from binary comparatives like preference connectives, e.g., $p > q$ for 'p is preferred to q' [16].

Despite the popularity of Dung's framework, it seems that the logical relations among attack statements have not been studied yet. Moreover, in abstract argumentation the notion of a set of arguments defending another argument has been defined. Again the logical relations among defend statements, and their relation to attack statements, seems unexplored. However, such an analysis would be useful for several reasons. It would give insight in Dung's abstract argumentation, it would be a basis for generalizations of Dung's theory, it would enable a comparison with other formalisms, and it would support reasoning about arguments in meta-argumentation [18]. We therefore raise the following questions in this paper:

1. What is a logic for abstract argumentation?
2. What are logical properties of abstract argumentation?
3. How to use such a logic to generalize Dung?
4. How is it related to conditional and preference logics?
5. How can agents reason about arguments?

Following Besnard and Doutre [1], to study these questions we represent arguments by propositions, such that "argument a together with argument b attacks argument c" is represented by $a \wedge b \rhd c$. We introduce connective '$\oslash$' for defence, so "argument a defends argument b" is represented by $a \oslash b$. Moreover, in the relation with conditional logic, we pursue the intuition that "argument a attacks argument b" is related to "if a holds then b does not hold". We relate, e.g.,

- if a attacks b and c defends b, then c attacks a,

to the following inference:

- from $a \rightarrow_{\rhd} \neg b$ and $c \rightarrow_{\oslash} b$, derive $c \rightarrow_{\rhd} \neg a$.

At present, this relation is not only unknown, but the question could not be raised, because there was no language in which it could be expressed.

Finally, reasoning about arguments in meta-argumentation is illustrated by the following dialogue:

A: I think arguments a and b defend argument c.
B: But argument d attacks argument c!
A: No problem, since argument a attacks argument d.

This dialogue illustrates how our logic contributes also to traditional argumentation theory.

The layout of this paper follows the research questions. In Section 2 we introduce a logical framework to reason about abstract argumentation. In Section 3 we consider logical properties among attack and defend statements, and in Section 4 we introduce a modal generalization of the logic to define variants and generalizations of Dung's theory. In Section 5 we consider the relation between the logic and more traditional formalisms, and in Section 6 we consider reasoning about arguments.

2 Semantics

We start with Dung's theory of argumentation. It is nowadays called a theory of abstract argumentation, because it ignores the internal structure of arguments. Here we use the presentation of Besnard and Doutre [1], who in contrast to Dung also define sets of arguments attacking other sets of arguments. Moreover, they write "argument system" where Dung writes "argument framework".

Definition 1 (Argument System). *An argument system is a pair $\langle A, R\rangle$, where A is a set (of arguments), and R is a binary relation over A which represents a notion of* attack *between arguments ($R \subseteq A \times A$). Given two arguments a and b, $(a, b) \in R$ means that a attacks b or that a is an attacker of b. A set of arguments S attacks an argument a if a is attacked by an argument of S. A set of arguments S attacks a set of arguments S' if there is an argument $a \in S$ which attacks an argument $b \in S'$.*

Dung assumes an argument system $\langle A, R\rangle$ to be given. Moreover, he gives several semantics which produce none, one or several sets of acceptable arguments called extensions. Most of these semantics depend on an additional notion of what is nowadays called defence. Instead of "S defends a", Dung says "a is acceptable with respect to S." We also define a set of arguments defending another set of arguments.

Definition 2 (Argument Semantics). *Let $\langle A, R\rangle$ be an argument system.*

- *$S \subseteq A$ is conflict free iff there are no a and b in S such that a attacks b.*
- *A conflict free set $S \subseteq A$ is a stable extension iff for each argument which is not in S, there exists an argument in S that attacks it.*
- *An argument $a \in A$ is defended by a set $S \subseteq A$ (or S defends a) iff for any argument $b \in A$, if b attacks a, then S attacks b.*
- *A conflict free set $S \subseteq A$ is admissible iff each argument in S is defended by S.*
- *A preferred extension is an admissible subset of A, which is maximal w.r.t. set inclusion.*
- *An admissible $S \subseteq A$ is a complete extension iff each argument which is defended by S is in S.*
- *The least (with respect to set inclusion) complete extension is the grounded extension.*

We say that $S \subseteq A$ defends $S' \subseteq A$ iff S defends each $a \in S'$.

The basic idea of the logic of abstract argumentation is that there is no longer a fixed argument system, in the following sense. A model of the logic represents an argument system, and such a model satisfies formulas representing that arguments attack or defend each other, or whether sets of arguments are extensions. Now, a formula is a theorem if it holds in all models, i.e., when it is true for every argument system. Theorems thus quantify over argument systems.

There are many ways to design a logic of abstract argumentation. In this section we stay close to Dung's argument system, and we generalize it in Section 4. We first assume a fixed signature or alphabet, which consists of the set of arguments A. L_0 is the set of conjunctions of atoms, representing sets of arguments, and L is the language that contains the notions of Dung's theory of argumentation. L_1 is the fragment of L that contains only the attack and defend connectives. Note that modalities in L cannot be nested.

Definition 3 (LAA language). *Given a set of arguments $A = \{a_1, \ldots, a_n\}$, we define the set L_0 of argument sets and the set L of LAA formulas as follows.*

L_0: $a_i \mid p \wedge q$ $\qquad (p, q \in L_0)$
L: $(p \rhd q) \mid (p \oslash q) \mid F(p) \mid S(p) \mid A(p) \mid P(p) \mid C(p) \mid G(p) \mid \neg\phi \mid (\phi \wedge \psi)$
$(p, q \in L_0; \phi, \psi \in L)$

We write L_1 for the fragment of L that does not contain a monadic modal operator. Moreover, disjunction $\vee$, material implication $\supset$ and equivalence $\leftrightarrow$ are defined as usual. We abbreviate formulas using the the following order on logical connectives: $\neg \mid \vee, \wedge \mid \rhd, \oslash \mid \supset, \leftrightarrow$. For example, $\neg p \rhd q \wedge r$ is short for $(\neg p \rhd (q \wedge r))$.

A semantic structure just consists of the binary attack relation R.

Definition 4 (LAA semantics). *Let A be set of arguments, let p and q be elements of L_0 and let ϕ and ψ be elements of L, and let R be a binary relation over A. We have:*

$R \models p \rhd q$ *iff in argument system $\langle A, R\rangle$, the set of arguments in p attack the set of arguments in q.*
$R \models p \oslash q$ *iff in argument system $\langle A, R\rangle$, the set of arguments in p defend the set of arguments in q.*
$R \models F(p)$ *iff the set of arguments in p is conflict free in argument system $\langle A, R\rangle$.*
$R \models S(p)$ *iff the set of arguments in p is a stable extension in argument system $\langle A, R\rangle$.*
$R \models A(p)$ *iff the set of arguments in p is admissable in argument system $\langle A, R\rangle$.*
$R \models P(p)$ *iff the set of arguments in p is a preferred extension in arg. system $\langle A, R\rangle$.*
$R \models C(p)$ *iff the set of arguments in p is a complete extension in arg. system $\langle A, R\rangle$.*
$R \models G(p)$ *iff the set of arguments in p is a grounded extension in arg. system $\langle A, R\rangle$.*
$R \models \neg\phi$ *iff not $R \models \phi$.*
$R \models \phi \wedge \psi$ *iff $R \models \phi$ and $R \models \psi$.*

Moreover, logical notions are defined as usual, in particular:

- $R \models \{\phi_1, \ldots, \phi_n\}$ *iff $R \models \phi_i$ for $1 \leq i \leq n$,*
- $\models \phi$ *iff for all R, we have $R \models \phi$,*
- $S \models \phi$ *iff for all R such that $R \models S$, we have $R \models \phi$.*

In this paper we are in particular interested in logic L_1 that only contains the attack and defend connectives, which constitute the basis of Dung's theory. We believe that to understand Dung's theory, one has first to better understand these two binary connectives.

Example 1. If a attacks b and c defends b, then c attacks a,

- $\models (a \rhd b) \wedge (c \oslash b) \supset (c \rhd a)$.

3 Logical Properties

The logical relations among attack formulas are characterized by the left (LD) and right distribution (RD) properties. They follow from the definition of attack among sets of arguments in terms of attacks among individual arguments. To understand this characterization we consider two logical consequences. First, logical consequences of the distribution properties (read from right to left) are left (LS) and right strengthening (RS). Right strengthening indicates that the attack connective does not behave like a conditional connective, but it behaves in this respect like a comparative connective (see Section 5 for details). Secondly, the more remarkable logical consequences of the distribution properties (read from left to right) is that if two arguments together attack another argument, then one of these arguments individually attacks the other argument (LT and RT). These splitting properties indicate room for generalizing Dung's theory (see Section 4).

$$\begin{array}{lll}
\mathrm{LD} \models (a \wedge b \rhd c) \leftrightarrow (a \rhd c) \vee (b \rhd c) & \mathrm{LA} & \models (a \oslash c) \vee (b \oslash c) \supset (a \wedge b \oslash c) \\
\mathrm{RD} \models (a \rhd b \wedge c) \leftrightarrow (a \rhd b) \vee (a \rhd c) & \mathrm{RD} & \models (a \oslash b \wedge c) \leftrightarrow (a \oslash b) \wedge (a \oslash c) \\
\mathrm{LS} \;\, \models (a \rhd c) \supset (a \wedge b \rhd c) & \mathrm{LS} & \models (a \oslash c) \supset (a \wedge b \oslash c) \\
\mathrm{RS} \;\, \models (a \rhd b) \supset (a \rhd b \wedge c) & \mathrm{RW} & \models (a \oslash b \wedge c) \supset (a \oslash b) \\
\mathrm{LT} \;\, \models (a \wedge b \rhd c) \supset (a \rhd c) \vee (b \rhd c) & \mathrm{RC} & \models (a \oslash b) \wedge (a \oslash c) \supset (a \oslash b \wedge c) \\
\mathrm{RT} \;\, \models (a \rhd b \wedge c) \supset (a \rhd b) \vee (a \rhd c) & &
\end{array}$$

The logical relations among defend relations are characterized by left additivity (LA) and right distribution (RD) properties. These properties follow from the definition of defend among sets of arguments in terms of attacks among individual arguments. The first logical consequences of these two properties (read from left to right) are left strengthening (LS) and right weakening (RW). Right weakening indicates that the defend connective behaves like a conditional connective (see Section 5 for details). Secondly, we have the conjunction property RC (read from right to left).

The relation among attack and defence connectives is as follows. If a set of arguments is finite, we can simply define the defend connective in terms of attack connective.

- $(a \oslash b) \leftrightarrow \bigwedge_{c \in A}((c \rhd b) \supset (a \rhd c))$

An instance of this relation, which characterizes the infinite case, is the following property already observed in Example 1. It says that the only possible defence is a direct counterattack, and thus rules out other defence tactics. This may seem counterintuitive at first sight, but it makes Dung's system effective.

- $(a \oslash b) \wedge (c \rhd b) \supset (a \rhd c)$

Though we are primarily interested in the logic L_1, the following example illustrates that the logic can be used to express well known relations among extensions.

Example 2. A stable extension is also a preferred extension, and a preferred extension is also a complete extension.

- $\models S(p) \supset P(p)$
- $\models P(p) \supset C(p)$

Two important properties are the expressive power of the language, and compactness of the logic.

Proposition 1 (Expressive power). *The logical language is expressive enough to distinguish two distinct argumentation theories based on the same set of arguments.*

Proof. *If two argumentation systems are distinct, then there are two arguments a and b such that $R_1(a, b)$ holds in one argument system $\langle A, R_1\rangle$, but $R_2(a, b)$ does not hold in the other $\langle A, R_2\rangle$ – or vice versa. Then we have $R_1 \models a \rhd b$, but not $R_2 \models a \rhd b$ – or vice versa.*

Proposition 2 (Compactness). *The logic is not compact, when the set of arguments A is infinite.*

Proof. *Follows directly from universal quantification in the definition of the semantics. For example, assume that A is infinite. We can derive that argument a defends argument b when there is an infinite set of formulas for each argument $c \in A$ that either a attacks c or c does not attack b. However, we cannot derive that a defends b from a finite set of formulas.*

A non-monotonic extension can be defined based on distinguished models and subset minimal attack relations. Sometimes such distinguished models are called preferred models and non-monotonic entailment is called preferential entailment.

Definition 5. *A model R is a distinguished model of a set of sentences S iff*

1. *$R \models S$, and*
2. *there is no $R' \subset R$ such that $R' \models S$.*

Nonmonotonic entailment is defined as usual:

- *$T \mid\!\sim \phi$ iff for all distinguished models R of T we have $R \models \phi$.*

The typical use of our logic is when an argument system is specified by a set of attack statements; we call such a set an argument specification.

Proposition 3. *An argument specification is a set of attack formulas $AS = \{p_1 \rhd q_1, \ldots, p_n \rhd q_n\}$. The distinguished model of an argument specification AS is unique.*

There are some limitations to the logic proposed here. First, the semantics leave little room for generalizations of Dung's theory. Secondly, we cannot express the characterizations in propositional logic provided by Besnard and Doutre. Thirdly, we cannot express that stable, preferred and complete semantics admit multiple extensions whereas the grounded semantics ascribes a single extension to a given argument system. In the following section we therefore discuss an extension of LAA in modal logic.

4 Modal Logic of Abstract Argumentation

To define variants and generalizations of Dung's theory, we now generalize LAA in a modal logic setting. We restrict ourselves to finite sets of arguments. Since sentences of the logic are finite, we cannot represent and reason about infinite extensions. The logic therefore seems most suitable for finite argument systems.

Our generalization is based on an attack relation between sets of arguments. Such sets of arguments are called positions and represented in the semantics of the logic by worlds in a possible worlds model. The attack relation is thus a binary relation between worlds, that is, a standard accessibility relation of possible worlds semantics.

Our motivation is that Dung's assumption that the attack relation exists between individuals arguments instead of sets of arguments is quite strong, and that it is not warranted in cases where the cumulative weight of arguments is decisive [15, 2]. For example, in some legal cases circumstantial evidence may be used in a cumulative way. Each piece of evidence individually would not be enough to connect a suspect to the crime scene, but many pieces of evidence taken together would be enough to conclude that the suspect was present at the crime scene. So only a set of arguments taken together would attack a position in this case.

Formally, we define a normal bimodal semantics in which modal operator $\Box_1$ represents the attack relation, and $\Box_2$ is a universal modality used for technical reasons. Since we have right strengthening for attack connectives where normal modal operators have right weakening, we use a negation in the definition of the attack connective. Propositional formulas represent positions, i.e., sets of arguments. The logic also has negations and disjunctions in the left and right hand side of our connectives, but we do not use this in this paper. We adapt the definition of defend in terms of attack to deal with our generalized setting (s represents a set of atoms as well as a conjunction of atoms).

Definition 6 (MLAA language). *Given a set of arguments $A = \{a_1, \ldots, a_n\}$, we define the set ML of MLAA formulas as follows.*

ML: $a_i \mid \Box_1(\phi) \mid \Box_2(\phi) \mid F(\phi) \mid S(\phi) \mid A(\phi) \mid P(\phi) \mid C(\phi) \mid G(\phi) \mid \neg\phi \mid (\phi \wedge \psi)$ $(\phi, \psi \in ML)$.

We write ML_1 for the fragment of ML that contains only monadic modal operators $\Box_1$ and $\Box_2$. Moreover, disjunction $\vee$, material implication $\supset$ and equivalence $\leftrightarrow$ are defined as usual. We extend the modal logic with the definition:

- $p \rhd q = \Box_2(p \supset \Box_1 \neg q)$
- $p \oslash q = \bigwedge_{s \subseteq A}(s \rhd q \supset p \rhd s)$

We abbreviate formulas using the the following order on logical connectives: $\neg \mid \vee, \wedge \mid \rhd, \oslash \mid \supset, \leftrightarrow$.

For space reasons we only introduce a semantics for ML_1. The other modalities can be described by a non-normal modal semantics only, as they do not satisfy weakening nor strengthening. From a logical point of view, MLAA is a standard normal modal logic with universal relation. Complexity and axiomatization of this logic are well known, see for example [8].

Definition 7 (MLAA semantics). *Let A be a set of arguments. A possible worlds model M is a structure $\langle W, R, V\rangle$ where W is a set (of worlds), R is a binary (attack) relation on W, and V is a valuation function which assigns a subset of A to each element of W.*

- *$M, w \models a$ iff $a \in V(w)$ for all arguments $a \in A$*
- *$M, w \models \neg\phi$ iff not $M, w \models \phi$*
- *$M, w \models \phi \wedge \psi$ iff $M \models \phi$ and $M \models \psi$*
- *$M, w \models \Box_1\phi$ iff for all w' such that $R(w, w')$ we have $M, w' \models \phi$.*
- *$M, w \models \Box_2\phi$ if for all $w \in W$, we have $M, w \models \phi$.*

We assume that W contains exactly one world for each subset of A.

Clearly, the language L is a fragment of ML. Moreover, Dung's theory can be characterized by the properties we already discussed:

- $\models (a \wedge b \rhd c) \leftrightarrow (a \rhd c) \vee (b \rhd c)$
- $\models (a \rhd b \wedge c) \leftrightarrow (a \rhd b) \vee (a \rhd c)$

We also consider some instances of Dung's theory. As far as we know, there is no systematic study of the possible instances of Dung's theory. We consider additional axioms we can impose on the logic MLAA. The first property we consider is irreflexivity of R, which corresponds to the property that no argument can attack itself:

IR $\neg(a \rhd a)$

The second property we consider is symmetry of the attack relation, which corresponds to the property that if argument a attacks argument b, then argument b attacks argument a.

S $(a \rhd b) \leftrightarrow (b \rhd a)$

Symmetry is not accepted often, because a counterexample attacks a general rule, but a general rule does not necessarily attack a counterexample. E.g., if Swans are white (a), but in Australia they found black swans (b) then we have $b \rhd a$ without $a \rhd b$. If the attack relation is symmetric, then the defend relation becomes reflexive, that is, each argument defends itself: $a \oslash a$.

Note that when we take traditional properties of conditional logic, we do not seem to get something useful. In particular, reflexivity (R) does not hold. Transitivity (T) means that if argument a attacks argument b, and argument b attacks argument c, then argument a should attack argument c. This does not hold either. Take $a = c$ for example, then we get $a \rhd a$, which conflicts with IR.

R $a \rhd a$
T $(a \rhd b) \wedge (b \rhd c) \supset (a \rhd c)$

Finally, if we would add accessibility relations for the monadic modal operators, then we can deal with the remaining problems observed at the end of Section 3.

Example 3. Grounded extension is unique:

- $\models G(p) \wedge G(q) \supset \Box_2(p \leftrightarrow q)$

Characterization of conflict free sets based on satisfiability checking condition of [1]:

- $C(p) \wedge (p \rhd q) \supset \Diamond_2(p \wedge \neg q)$

5 Relation with Traditional Formal Systems

5.1 Preference Logic

Since the logic of the attack connectives satisfies left and right strengthening, it seems that it may be related to preference logic. In particular, “argument a attacks argument b” may be interpreted as “argument a is preferred to argument b”. However, this is less helpful than it may seem at first sight, because the area of preference logic is characterized by lack of consensus. In this subsection we make some observations.

First, the most popular branch of preference logic, as initiated in the sixties by Von Wright [16], is concerned with ceteris paribus preferences. This means that $p > q$ is interpreted as a preference of p over q all else being equal, typically interpreted as ‘under the same (or similar) circumstances’. When we consider a language that does not contain disjunction or negation, then such preferences are characterized by the following property of simultaneous left and right strengthening. This is strictly weaker than left and right strengthening of attack connectives considered in this paper.

- $p > q \supset p \wedge r > q \wedge r$

Secondly, in preference logic left and right strengthening are properties of so-called strong preferences, whereas so-called weak preferences do not satisfy left and right strengthening. The typical example of a strong preference $p > q$ is interpreted as “all $p \wedge \neg q$ worlds are strictly preferred to all $q \wedge \neg p$ worlds”. Without a ceteris paribus proviso these strong preferences are known to be too strong to be useful in practice, because for example $p > \neg p$ together with $q > \neg q$ is inconsistent. The reason is that such strong preferences also satisfy the following property of asymmetry. However, for the attack connective we can easily have that argument a attacks argument b, while at the same time argument b attacks argument a.

- $p > q \rightarrow \neg(q > p)$.

Thirdly, the attack relation behaves like a non-strict preference interpreted on a partial pre-order and defined by “$p \geq q$ iff there is no $q \wedge \neg p$ world that is strictly preferred to a $p \wedge \neg q$ world”. As far as we can see at this moment, this relation seems coincidental and does not seem to refer to any deep connection between the two logical systems.

5.2 Conditional Logic

The defend connective behaves like a standard conditional connective, with one important exception: it does not satisfy the identity rule. An argument a does not necessarily defend argument a, because when another argument b attacks argument a, there is no reason why argument a attacks argument b (unless the attack relation is symmetric, of course).

Consequently, to consider the defend connective we need an identity free logic, which are rare. Here we use input/output logic [10, 11], which has been proposed in philosophical logic for normative or deontic reasoning, and which have been used in artificial intelligence to characterize causal reasoning [3] and logic programming [4].

To emphasize the lack of identity, Makinson and van der Torre write their conditional "if input a, then output x" as (a, x).

The defend connective behaves like so-called simple-minded output, which is defined as a closure on a set of conditionals under replacements of logical equivalents, and the following three proof rules of strengthening of the input, weakening of the output, and the conjunction rule for the output. See the above mentioned papers for semantics of this proof system.

Definition 8. *Let AS be a set of defend formulas $\{p_1 \oslash q_1, \ldots, p_n \oslash q_n\}$. Simple-minded output is the closure of AS under replacement of logical equivalents, and the following three rules.*

$$\frac{a \oslash x}{a \wedge b \oslash x} SI \qquad \frac{a \oslash x \wedge y}{a \oslash x} WO \qquad \frac{a \oslash x, a \oslash y}{a \oslash x \wedge y} AND$$

We can also formalize the attack relation as an input/output logic, if we use the same encoding as we used in modal logic MLAA, that is, if we add a negation before the output. This is done by representing "argument a attacks b" by "if input a, then output $\neg b$". Weakening of the output is then transformed into strengthening of the output, and the right conjunction rule is transformed into right disjunction. However, the latter rule is not meaningful as we have not defined disjunctions for argument sets.

Definition 9. *Let AS be an argument specification. Simple-minded output is the closure of AS under replacement of logical equivalents, and the following three rules.*

$$\frac{a \rhd x}{a \wedge b \rhd x} SI \qquad \frac{a \rhd x}{a \rhd x \wedge y} SO \qquad \frac{a \rhd x, a \rhd y}{a \rhd x \vee y} OR$$

At this point, it is very tempting to define both attack and defend in a single conditional logic to study their interaction. In other words, it is tempting to consider an input/output logic in which a conditional (p, q) is read as 'p defends q', and $(p, \neg q)$ is read as 'p attacks q', for p and q conjunctions of atomic propositions. This is formalized in the following definition of the input/output logic of abstract argumentation.

Definition 10. *Let IOLAA be simple minded output, together with the following two definitions for p and q conjunctions of atomic propositions.*

- $p \rhd q = (p, \neg q)$
- $p \oslash q = (p, q)$

Let us now consider the relation between attack and defend in IOLAA. The characteristic axiom that a defends b implies that if c attacks b, then a also attacks c, is given by the following unusual rule: $\frac{(a,b),(c,\neg b)}{(a,\neg c)}$. However, clearly we do not want to derive that a defends b implies that if c attacks b, then c also attacks a, that is: $\frac{(a,b),(c,\neg b)}{(c,\neg a)}$. Note that the distinction between these two inference rules is whether the formulas start with a negation symbol. Consequently we cannot accept one without the other, unless we add additional syntactic constraints. This illustrates that the formalization of argumentation theory in input/output logic needs further investigation.

6 Use in Argumentation

The typical approach in argumentation is that each participant makes an argument, and then argumentation theory is used to determine grounded, stable, or preferred sets of arguments. Reasoning of the agents occurs only at the level of constructing arguments, and the role of logic has been restricted to the internal structure of arguments.

A: I think p.
B: I think not p, because q.
A: But not q, because of r.

In the analysis of argumentation, reasoning about arguments has been restricted to meta-rules such as, for example, the order of arguments, or the choice of words. However, Dung [7] has shown that reasoning about arguments can also be based on concepts such as attack and defence. A typical example may be:

A: I think p and q defend r.
B: But s attacks r.
C: No problem, since p attacks s.

Note that the agents do not enumerate the complete argument system, that is, they do not list the complete attack relation R of the argument system $\langle A, R\rangle$. To formalize this example we therefor cannot assume a fixed argument system $\langle A, R\rangle$, as Dung does. We need the logical language to quantify over argument systems.

In this example, the agents make arguments like "p and q defend r" which themselves refer to arguments p, q and r. The former may therefore be called meta-arguments. The logic that formalizes or characterizes the reasoning of agents about arguments, when they construct meta-arguments, is therefore at first sight quite different from the logic typically used in argumentation. We therefore believe that the confinement of logic to the internal structure of arguments is too limited; there is also a role of logic in the formalization of reasoning about arguments.

We agree with Wooldridge *et al* that meta-argumentation is particularly useful for agent theory [18]. This meta-level could be used, potentially, to speed up argumentations by means of a kind of "caching" function. Just like in chess (Polish opening), you can use patterns of arguments, give them a name, and know that such a pattern attacks or defends another pattern. If you respect your opponent, there is no need to "play out" the whole argument.

Wooldridge *et al* [18] propose a hierarchical first-order meta-logic, which enables them to distinguish object level statements, arguments made about these object level statements, and statements about arguments. Such a distinction is commonly accepted in dialogue systems [13]. However, as a consequence their formal system appears to be more complex, and it is less clear how to relate their formal system to other formal approaches in the way we have related our system to reasoning about preferences or conditionals. A comparison between the two approaches is left for further research.

7 Concluding Remarks

In this paper we introduce a logic of abstract argumentation called LAA, with two properties usually not considered in formal theories of argumentation: it formalizes logical relations among attack and defend formulas, and it formalizes reasoning which does not assume a fixed argument system, but which quantifies over argument systems. We first define a logical system that is very close to Dung's theory. We show some properties of this logical system, focussing on the logical relations among attack and defend formulas. We also relate the logic and its properties to more traditional preference and conditional logics. To generalize Dung's setting, we turn the logic LAA into a normal bimodal logic MLAA. We define positions as sets of arguments, and define the attack relation as a binary relation between positions. We suggest that in reasoning about arguments, Dung's assumption that an argument system is given is too strong.

There are several issues for further research. For example, the modal logic represents negations and disjunctions in the left and right hand side of the attack and defend connectives. Can they be given a useful interpretation? When 'a' means that argument a has been made, then '$\neg a$' may mean that a is withdrawn in the sense that there is no longer a commitment to defend it. In such a setting, one may look into properties like:

$$\frac{a \rhd b \wedge c, a \rhd b \wedge \neg c}{a \rhd b} \qquad \frac{a \rhd b, \neg b \rhd c}{a \rhd c}$$

Moreover, the content level of argumentation contains much more than just propositional logic, including attacks and defend expressions. In a straightforward extension of our logic, reasoning about arguments such as "$a \rhd b$ attacks $c \oslash b$" can be studied using nested connectives:

$$(a \rhd b) \rhd (c \oslash b)$$

Reasoning becomes argumentation once there are two agents with opposing views. Such agents may have have beliefs and goals. Moreover, in a goal-based dialogue with sub-goals to achieve it, there may be dialogue fragments representing each of the subgoals, probably in the same order. The use and extension of our logic for such dialogues is another topic for further research.

The modal characterization outlined in the paper raises an interesting issue, which might be worth exploring in future research. At first sight it opens the door for the application of model-checking techniques, initially used for automatically verifying a Kripke structure (describing the execution of a program) against a number of 'correctness' requirements. It is natural to ask if such techniques can be applied to argumentation. The Kripke structure to be model-checked describes an argument system (or, if you wish, a dialogue) rather than the execution of a program. And the "correctness" requirement is expressed as a formula f in MLAA rather than a temporal formula. For instance, termination seems to be an essential property of both programs and dialogues.

Finally, Bochman [5] recently introduced a logic of propositional argumentation based on the assumption-based argumentation framework of Bondarenko *et al.* [6]. A comparison is left for further research.

References

1. P. Besnard and S. Doutre. Checking the acceptability of a set of arguments. In *Procs. of NMR04*, pages 59–64, 2004.
2. A. Bochman. Collective argumentation and disjunctive logic programming. *Journal of Logic and Computation*, 13:405–428, 2003.
3. A. Bochman. A causal approach to nonmonotonic reasoning. *Artificial Intelligence*, 160(1-2):105–143, 2004.
4. A. Bochman. A causal logic of logic programming. In *Procs. of KR 2004*, pages 427–437, 2004.
5. A. Bochman. Propositional argumentation and causal reasoning. In *Procs. of IJCAI95*, pages 388–393, 2005.
6. A. Bondarenko, P. Dung, R. Kowalski, and F. Toni. An abstract, argumentation based approach to default reasoning. *Artificial Intelligence*, 93(1-2):63 – 101, 1997.
7. P. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 76:321–358, 1995.
8. E. Hemaspaandra. The price of universality. *Notre Dame Journal of Formal Logic*, 37(2):174–203, 1996.
9. J. Horty. Argument construction and reinstatement in logics for defeasible reasoning. *Artificial Intelligence and Law*, 9:1–28, 2001.
10. D. Makinson and L. van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29:383–408, 2000.
11. D. Makinson and L. van der Torre. Constraints for input-output logics. *Journal of Philosophical Logic*, 30(2):155–185, 2001.
12. S. D. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
13. H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6:231–287, 1998.
14. H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Guenthner, editors, *Handbook of philosophical logic*, pages 218–319. Kluwer, Dordrecht, 2002.
15. B. Verheij. Accrual of arguments in defeasible argumentation. In *Proceedings of the Second Dutch/German Workshop on Nonmonotonic Reasoning*, pages 217–224, 1995.
16. Georg Henrik Von Wright. *The Logic of Preference: an Essay*. Edinburgh University Press, 1963.
17. C. Walton. Model checking agent dialogues. In *Proceedings of DALT'04*, LNCS, pages 132–147. Springer, 2005.
18. M. Wooldridge, P. McBurney, and S. Parsons. On the meta-logic of arguments. In *Proceedings of AAMAS'05*, pages 560–567, 2005.

On the Meta-logic of Arguments

Michael Wooldridge[1], Peter McBurney[1], and Simon Parsons[2]

[1] Dept of Computer Science
University of Liverpool
Liverpool L69 7ZF, UK
{mjw, peter}@csc.liv.ac.uk
[2] Dept of Computer and Information Science
Brooklyn College, CUNY
Brooklyn, 11210 NY, USA
parsons@sci.brooklyn.cuny.edu

Abstract. Argumentation has received steadily increasing attention in the multi-agent systems community over the past decade, with particular interest in the use of argument models from the informal logic community. The *formalisation* of such argument systems is a necessary step if they are to be successfully deployed, and their properties rigorously understood. However, there is as yet no widely accepted approach to the formalisation of argument systems. In this paper, we take as our starting point the view that arguments and dialogues are inherently *meta-logical*, and that any proper formalisation of argument must embrace this aspect of their nature. For example, a statement that serves as a justification of an argument is is statement *about* an argument: the argument for which the justification serves must itself be referred to in the justification. From this starting position, we develop a formalisation of arguments using a hierarchical first-order meta-logic, in which statements in successively higher tiers of the argumentation hierarchy refer to statements further down the hierarchy. This enables us to give a clean formal separation between object-level statements, arguments made about these object level statements, and statements about arguments.

1 Introduction

Argumentation has received steadily increasing attention in the multi-agent systems community over the past decade, with particular interest in the use of argument models from the informal logic community such as that of Walton and Krabbe [19, 24]. The *formalisation* of such argument systems is a necessary step if they are to be successfully deployed, and their properties rigorously understood. Most argument systems can be classified according to whether the arguments they consider are structured, typically logical entities (e.g., [2, 11, 12, 13]), or atomic, abstract entities (in the sense of Dung's abstract argument model [7, 1]). However, although some research has considered the links between these different types of systems [3], no one model is universally accepted, and both the abstract and logical argumentation paradigms have well-known problems as a model of rational argument [18].

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 42–56, 2006.

In this paper, we focus on a logic-based view of arguments [13]. We take as our starting point the view that arguments and dialogues are inherently *meta-logical* processes. By this, we mean that the arguments made by protagonists in a debate must *refer* to each other. This is because arguments are not just about which states of affairs exist in the world, or how objects in the world stand in relation to one-another. If this were the case, then dialogues would be impoverished indeed, essentially restricted to asserting the truth or falsity of statements. We believe that rational argumentation also involves putting forward *arguments about arguments*, and it is in this sense that they are *meta-logical.* For example, a statement that serves as a justification of an argument is a statement *about* an argument: the argument for which the justification serves must itself be *referred to* in the justification.

One of our main aims in this paper is to put this idea of meta-argument on the map of argumentation research. But we also hope to show how a meta-logical treatment of argument can clarify some apparently difficult issues in the formalisation of argument. Our basic approach involves developing a *hierarchical* formalisation of logic-based arguments. That is, we construct a (well-founded) tower Δ_0, Δ_1, ... of arguments, where arguments, statements, and positions at a level n in the hierarchy may refer to arguments and statements at levels m, for $0 \leq m < n$. In the bottom tier Δ_0 of the hierarchy are *object level* statements about the domain of discourse. The apparatus we use for formalising such an argument system is a *hierarchical first-order meta-logic*, a type of first-order logic in which individual terms in the logic can refer to terms in another language (cf. Konolige's first-order formalisation of knowledge and action [10]). This formalisation enables us to give a clean formal separation between object-level statements, arguments made about these object level statements, and statements about arguments.

The remainder of the paper is structured as follows. First, in the following section, we give a motivation and informal introduction to the framework. In section 3, we present a proof-of-concept formalisation of our approach using hierarchical meta-logic, and in section 4, we present some conclusions. Our work makes two key contributions to the theory of argumentation. First, and perhaps most importantly, we motivate and establish the notion of meta-argumentation as an issue in its own right, and present a first formalisation of this process. Although meta-*languages* have been used in the formalisation of dialectical systems [20], to the best of our knowledge we are the first to use a *meta-logic* in this way. Our second contribution is to show how a number of different approaches to argumentation may be uniformly combined within the meta-logic framework: in particular, the logic-based approaches of [2, 13], the abstract argumentation framework of Dung [7], and Bench-Capon's value-based argumentation framework [1]. Note that the integration of abstract argument frameworks and logic-based frameworks is possible only *because* we adopt a meta-logical perspective: the integration involves stating and reasoning about relations over logical formulae, which cannot be achieved without some meta-logical apparatus.

2 A Hierarchical System of Arguments

Before proceeding to the formal details of our approach, we present some more detailed motivation for it. As noted in the introduction, our key motivation is the following observation:

Argumentation and formal dialogue is
necessarily a meta-logical process. (*)

This seems incontrovertible: even the most superficial study of argumentation and formal dialogue indicates that, not only are arguments made about object-level statements, they are also made *about* arguments. In such cases, an argument is made which *refers* to another argument. Moreover, there are clearly also cases where the level of referral goes even deeper: where arguments refer to arguments that refer to arguments. Perhaps the paradigm examples of such meta-argumentation would be in a courtroom setting, where an advocate objects to an argument of the opposing advocate, or where a judge rules an argument inadmissible. Here, the arguments being put forward refer to arguments made about the domain of discourse, but are clearly not actually about the domain of discourse itself.

If one accepts the validity of (*), then it is natural to view argument as taking place at a number of levels. At the lowest level, we do not really have arguments at all – we have statements about the domain of discourse. At the next level in the argumentation hierarchy, we have arguments themselves: these are statements about the object-level statements, and so on. Of course, in any attempt to formalise such a model of arguments, we must define the composition of each level of the hierarchy. There are many choices to be made here – particularly at higher levels of the hierarchy – and we are in no position to give a canonical view. In this paper, we set out and work with a 3-tier hierarchy, as illustrated in Figure 1. Throughout the remainder of the paper, we will denote these levels of the hierarchy by Δ_0, Δ_1, etc., with Δ_0 always being the *lowest* level of the hierarchy. The tiers of the hierarchy are as follows:

Δ_0 *The Object Level:* This tier of the hierarchy does not actually contain arguments at all. It consists of statements about the domain of discourse, and in particular defines the interrelationships between the entities in the domain of discourse. In a legal setting (which is perhaps the paradigm example of a domain for formal argument and discourse), we can think of Δ_0 as consisting of the established facts of the case, (such as evidence that may be introduced), as well as non-logical axioms about the domain.

Δ_1 *Ground Arguments:* Arguments exist for the first time as first class entities in this tier of the hierarchy. Δ_1 defines what constitutes an argument: in the model of argument that we use, an argument consists of a conclusion and some supporting statements, with a notion of logical consequence between them [2, 13]. By contrast, in Toulmin's scheme an argument is more complex, consisting of a claim (e.g., "John is old") , a warrant (e.g., "over 70 is old") with associated backing (e.g., some demographic data), and some data (e.g., "John is 78") [22]. Note that the hierarchical meta-logic approach itself is consistent with both such models of argument, and indeed many others; but we find it convenient to work with the logical model. Since we can refer to arguments in this tier in the hierarchy, we can also capture relationships *between* arguments here. For example, the canonical notion of one argument *attacking* another is a relation between arguments [7], and cannot therefore be present at any lower tier of the hierarchy. Although "attack" is one relation that may exist between arguments, it is of course not the only one: since the object level Δ_0 will often contain inconsistencies, the notion of attack will often not be enough to obtain a useful coherent view. We therefore use Bench-Capon's notion of *value*-based argument, which overlays attack with *values* that the argument appeals to, and hence makes it possible to choose between arguments on the basis of the values they represent [1].

Δ_2 *Meta-Arguments:* Notice that at the Δ_1 tier of the hierarchy, we can make statements that are about object-level statements, (e.g., we can assert that a particular structure represents an acceptable argument) but we cannot directly refer to the *process* by which an argument is established. That is, in Δ_1 we cannot say that "we can establish that a is an argument using axioms T". Hence properties of arguments that involve referring to the axioms or procedures via which we in fact establish that they are arguments cannot be captured in Δ_1. However, such properties *can* be captured in Δ_2. In particular, the main construction used in Δ_2 is that of an argument referring to an argument. To illustrate the value of this, we will show how we can distinguish in Δ_2 between "classical" $\mathcal{L}_1$ arguments (in which the full technical apparatus of classical logic proof can be used to establish a conclusion), and *intuitionistic* $\mathcal{L}_1$ arguments, where a more restrained (and some would argue more realistic) notion of proof is used [6].

Of course, there is no reason why this hierarchy should not be continued: the same logical apparatus we use can essentially be copied into layers further up the hierarchy, permitting arguments about arguments about arguments ... as desired. Where argumentation is used in human settings, this is exactly what seems to happen: consider an argument that takes place between advocates in a court of law, and then (further up the hierarchy), arguments made by the judge about these arguments, and then potentially arguments made in a supreme court about the arguments made by the judge in the lower court. To cleanly (and properly) capture this kind of setting, it seems to us that our hierarchical approach is not only appropriate, but perhaps essential. However, for the purposes of this paper, we will restrict our attention to the three layers indicated here.

3 The Formal Framework

Meta-level reasoning (reasoning about reasoning) has a venerable history in artificial intelligence, and *logical* approaches to meta-level reasoning have been widely studied, with a range of approaches developed and evaluated (see [8, pp.239–262] and [16, 5] for reviews). For our purposes, the most suitable

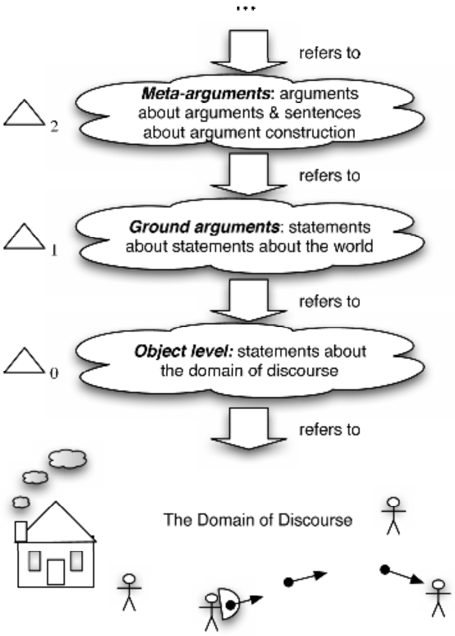

...
refers to
$\triangle_2$
Meta-arguments: arguments about arguments & sentences about argument construction
refers to
$\triangle_1$
Ground arguments: statements about statements about the world
refers to
$\triangle_0$
Object level: statements about the domain of discourse
refers to
The Domain of Discourse

Fig. 1. A hierarchy of arguments

formalism to adopt is a *first-order meta-logic* [8, Chapter 10]. Viewed in the most abstract way, a first-order meta-logic is simply a first-order logic whose domain (the set of entities that may be referred to in the language) includes sentences of another language (the *object* language). An important distinction is made between meta-languages that can refer to themselves (i.e., languages whose domain contains the set of sentences of the language itself), which are usually called *self-referential*, and those where this is not possible. Self-referential languages tend to be rather complex and intricate systems to deal with: first because when one assumes even seemingly innocent and innocuous axioms they tend to become inconsistent, and second because they allow one to express paradoxical statements such as the "liar" paradox [15, 23]. *First-order hierarchical meta-languages* provide a somewhat more stable logical foundation [23]. The basic idea of such languages is that we define a (well-founded) tower of languages $\mathcal{L}_0 - \mathcal{L}_1 - \cdots$, such that the domain of $\mathcal{L}_0$ is the set of entities in the domain of discourse, and the domain of each language $\mathcal{L}_u$ for $u > 0$ contains the set of formulae of language $\mathcal{L}_v$ for $0 \leq v < u$, but contains no sentences from languages $\mathcal{L}_w$ for $w \geq u$. In this way, we have the ability to make statements about statements about statements ... to some arbitrary level of depth, but because our languages are strictly hierarchical (can only refer to sentences of languages further down the hierarchy), self-reference (and all the logical problems it entails) is not possible. Hierarchical meta-languages have been used as the basis of several formalisms for reasoning about action (see, e.g., [10]) and recently the approach of using meta-language predicates in place of modal operators for referring to (for example) what an agent knows or believes has undergone something of a revival (see, e.g., [9]). We will present our formalisation of each tier in the hierarchy in turn, starting with $\varDelta_0$.

Note that we do not give a syntax and semantics for each language $\mathcal{L}_i$, as these are available elsewhere in the literature (e.g., [10, 23]). We will assume that the languages contain the conventional logical connectives of negation ("$\neg$"), disjunction ("$\vee$"), conjunction ("$\wedge$"), implication ("$\rightarrow$"), and bi-conditional ("$\leftrightarrow$"), the usual apparatus of first-order quantification ("$\forall$" "$\exists$"), functional terms, equality, and logical constants for truth ("**true**") and falsity ("**false**"). Moreover, for each language $\mathcal{L}_i$ we assume a logical consequence relation $\models_{\mathcal{L}_i}$. Technically, each level $\varDelta_i$ in our hierarchy will constitute a *theory* in the language $\mathcal{L}_i$.

3.1 The Object/Domain Level: $\boldsymbol{\varDelta_0}$

We can understand $\varDelta_0$ as stating the basic "facts" of the argumentation domain, and the non-logical axioms associated with it[1]. We often refer to $\varDelta_0$ as the object-level, or domain theory. Thus, in the domain theory $\varDelta_0$, we define all the properties about the argumentation domain that may be admitted into the discourse. For simplicity of exposition here, we will assume that these are expressed using propositional logic, although of course there is no reason in principle why one should not use a richer language. Formally, $\varDelta_0$ will be a set of formula expressed in propositional logic.

Example 1. Here is an example domain theory:

$$\varDelta_0 = \{p, t, p \rightarrow q, (q \vee r) \rightarrow \neg s, t \rightarrow \neg p\}.$$

3.2 Arguments About the Domain: $\boldsymbol{\varDelta_1}$

Let us now move one step up the hierarchy. At level 1 in the hierarchy, we define our basic model of arguments: what constitutes an acceptable argument according to the underlying system of argument that we are interested in. In line with [2, 13], we consider an argument with respect to a domain theory $\varDelta_0$ as a pair $\langle \varphi, \varGamma \rangle$, such that:

1. $\varphi \in \varDelta_0$ is an $\mathcal{L}_0$-formula known as the *conclusion* of the argument and $\varGamma \subseteq \varDelta_0$ is a set of $\mathcal{L}_0$-formulae known as the *support*;
2. $\varGamma$ is consistent (i.e., not $\varGamma \models_{\mathcal{L}_0}$ **false**);
3. φ logically follows from $\varGamma$ (i.e., $\varGamma \models_{\mathcal{L}_0} \varphi$); and
4. there is no subset $\varGamma'$ of $\varGamma$ satisfying (2) and (3).

We now formalise this in our hierarchical logic framework. We must first put in place some conventions. First, recall that the domain of language $\mathcal{L}_1$ contains the expressions of $\mathcal{L}_0$. We assume that, for each primitive $\mathcal{L}_0$ expression e, there

[1] By "non-logical" axioms, we mean axioms or rules which refer specifically to the domain at hand, and which are not valid according to the semantics of the logic.

is a corresponding $\mathcal{L}_0$ term e'. $\mathcal{L}_0$ terms denoting compound object-language formulae are constructed using the meta-language functions *and*, *or*, *not*, and so on. Thus *or* is an $\mathcal{L}_1$ functional term which takes two arguments, each of which is an $\mathcal{L}_1$ term denoting an $\mathcal{L}_0$ formula: the function returns the $\mathcal{L}_0$ sentence corresponding to the disjunction of its arguments. For example, the $\mathcal{L}_0$ formula

$$p \rightarrow (q \vee r)$$

is denoted by the $\mathcal{L}_0$ term

$$imp(p', or(q', r')).$$

Since this construction is somewhat cumbersome, we follow standard practice and use *sense quotes* (sometimes called Frege quotes or Gödel quotes) as abbreviations:

$$\begin{array}{ll} \ulcorner \neg p \urcorner & \hat{=}\ not(p') \\ \ulcorner p \vee q \urcorner & \hat{=}\ or(p', q') \\ \text{etc.} & \end{array}$$

We will also assume that we have terms in $\mathcal{L}_1$ that stand for *sets* of $\mathcal{L}_0$ formulae. To build sets formally, we use an $\mathcal{L}_1$ constant $\emptyset$, which denotes the empty set of $\mathcal{L}_0$ formulae, and unary function $set(f)$, which takes an $\mathcal{L}_1$ term denoting an $\mathcal{L}_0$ formula, and returns the singleton set of $\mathcal{L}_0$ formulae containing the formula denoted by f. Finally, we use a binary function $union(T_1, T_2)$, which takes as arguments two $\mathcal{L}_1$ terms, each of which denotes a set of $\mathcal{L}_0$ formulae, and returns the set of $\mathcal{L}_0$ formulae corresponding to the union of these two sets. To make this somewhat more readable, we will write

$$\{\ulcorner \varphi_1 \urcorner, \ulcorner \varphi_2 \urcorner, \ldots, \ulcorner \varphi_k \urcorner\}$$

as an abbreviation for the following, somewhat more cumbersome $\mathcal{L}_0$ term:

$$union(set(\ulcorner \varphi_1 \urcorner), union(set(\ulcorner \varphi_2 \urcorner), \ldots, set(\ulcorner \varphi_k \urcorner) \ldots))$$

Finally, if T is an $\mathcal{L}_1$ term that stands for a set of $\mathcal{L}_0$ formulae, and f is an $\mathcal{L}_1$ term that stands for an $\mathcal{L}_0$ formula, then we write $FACT_1(T, f)$ to indicate that the formula denoted by f is a member of the set denoted by T. Note that the subscript "1" in the name of the predicate is to give the reader some visual clues as to which language this predicate belongs to: that is, it belongs to $\mathcal{L}_1$. We will also use $FACT_n(\cdots)$ predicates further up the hierarchy. For every statement f appearing in the domain theory Δ_0, we need to include in Δ_1 that f is a $FACT_1(\cdots)$ of Δ_0.

$$FACT_1(\Delta_0, f) \quad \text{for each } f \in \Delta_0$$

The next step is to introduce a predicate $PRV_1(\cdots)$, for provability. This is a binary predicate, taking arguments denoting a set of $\mathcal{L}_0$ formulae and an $\mathcal{L}_0$ formula, with the intended interpretation that $PRV_1(T, f)$ means that the formula denoted by f is provable from the theory denoted by T. To ensure that the predicate behaves as intended, we give axioms in Δ_1 that correspond to provability in $\mathcal{L}_0$. So, for example, this axiomatization will include the following, which capture that any member of T is provable from T, two axioms characterising reduction ad absurdum, i.e., that $\neg\neg f \leftrightarrow f$[2], modus ponens, and that if $f \wedge g$ can be proved from T, then so can f and so can g. (Note: In these axioms, and the remainder of the paper, to make formulae more readable, we will adopt the convention that *free variables are assumed to be universally quantified.*)

$$\begin{array}{l}
FACT_1(T,f) \rightarrow PRV_1(T,f)\\
PRV_1(T,f) \rightarrow PRV_1(T,not(not(f)))\\
PRV_1(T,not(not(f))) \rightarrow PRV_1(T,f)\\
PRV_1(T,imp(f,g)) \wedge PRV_1(T,f) \rightarrow PRV_1(T,g)\\
PRV_1(T,and(f,g)) \rightarrow (PRV_1(T,f) \wedge PRV_1(T,g))\\
\text{etc.}
\end{array}$$

It is straightforward to extend these axioms to give an $\mathcal{L}_1$ axiomatization that characterises $\mathcal{L}_0$ provability: for simplicity, we assume a set of axioms that characterises a *complete* proof system for $\mathcal{L}_0$ (see, e.g., [8, pp.55–62]).

Of course, for different purposes, different types of proof may be appropriate in the characterisation of $PRV_1(\cdots)$. We can *tailor* our notion of $\mathcal{L}_0$ provability by choosing different axioms characterising $PRV_1(\cdots)$. For example, if (for some reason) we wanted a notion of provability that did not include the ability to apply the and-elimination rule, then we would omit the fifth axiom for $PRV_1(\cdots)$ from the list above; if we wanted a constructive, intuitionistic notion of proof, then we would give an axiomatization without the second and third axioms, and so on.

Next, we define the subset relation over sets of $\mathcal{L}_0$ formulae as follows.

$$\begin{array}{l}
(T_1 \subseteq T_2) \leftrightarrow\\
\quad \forall f \cdot FACT_1(T_1,f) \rightarrow FACT_1(T_2,f)
\end{array}$$

We now introduce arguments. We use an $\mathcal{L}_1$ function $\langle\cdots\rangle$ of two arguments, which simply makes a tuple out of these arguments; where a is an $\mathcal{L}_1$ term denoting an argument, we use the projection function $conc(a)$ to extract the conclusion from argument a, and $supp(a)$ to extract the support.

$$\begin{array}{l}
conc(\langle f, T\rangle) = f\\
supp(\langle f, T\rangle) = T
\end{array}$$

We then say that $\langle f, T\rangle$ is a *prima facie* argument if f is provable from T and T is a subset of Δ_0.

$$\begin{array}{l}
PF_1(a) \leftrightarrow\\
\quad (PRV_1(supp(a), conc(a)) \wedge (supp(a) \subseteq \Delta_0))
\end{array}$$

(Recall that Δ_0 here is an $\mathcal{L}_1$ constant which denotes the set of $\mathcal{L}_0$ formula characterising the object level domain of discourse.)

[2] Note that we could collapse these two axioms into one biconditional; the rationale for not doing this will become clear in the following section.

A consistent prima facie argument ($CPF_1(a)$) is one whose support is consistent;

$$\begin{array}{l} CPF_1(a) \leftrightarrow \\ \quad (PF_1(a) \wedge \neg PRV_1(supp(a), \ulcorner \mathbf{false} \urcorner)) \end{array}$$

And an argument is a consistent prima facie argument that is minimal, in the sense that no subset of the support is sufficient to serve as a support for the argument.

$$\begin{array}{l} ARG_1(a) \leftrightarrow \\ \quad (CPF_1(a) \wedge \neg \exists T \cdot (T \subseteq supp(a)) \wedge CPF_1(\langle conc(a), T \rangle)) \end{array}$$

Example 2. Suppose that Δ_0 is as defined in (Ex_1), above. Then, constructing Δ_1 using the axioms and facts as above, we can conclude the following.

$$\begin{array}{l} \Delta_1 \models_{\mathcal{L}_1} ARG_1(\langle \ulcorner q \urcorner, \{\ulcorner p \urcorner, \ulcorner p \rightarrow q \urcorner\} \rangle) \\ \Delta_1 \models_{\mathcal{L}_1} ARG_1(\langle \ulcorner \neg p \urcorner, \{\ulcorner t \urcorner, \ulcorner t \rightarrow \neg p \urcorner\} \rangle) \\ \Delta_1 \models_{\mathcal{L}_1} ARG_1(\langle \ulcorner \neg s \urcorner, \{\ulcorner p \urcorner, \ulcorner p \rightarrow q \urcorner, \ulcorner (q \vee r) \rightarrow \neg s \urcorner\} \rangle) \end{array}$$

We now formalise the way that arguments may *attack* one another [7]. In the argumentation literature, "a_1 attacks a_2" is roughly interpreted as meaning "a rational agent that accepts a_1 would have to reject a_2". Unfortunately, there is no consensus on the semantics of attacks, and indeed Dung's abstract argumentation theory completely ignores the issue, simply assuming that one is presented with an attack relation. In logic-based argument, there are two widely used notions of attack: *rebuttal* (where the conclusion of one argument is logically equivalent to the negation of the conclusion of the other) and *undercutting* (where the conclusion of one argument is logically equivalent to the negation of some element of the support): see, e.g., [17]. Since rebuttal is inherently symmetric (in the sense that if a_1 rebuts a_2, then by definition a_2 rebuts a_1), its value in the definition of attack has been questioned [2]. For this reason, we will focus on undercutting as the foundation of attack.

We define a two place $\mathcal{L}_1$ predicate $ATTACK_1(\cdots)$, such that $ATTACK_1(a_1, a_2)$ means that a_1 undercuts a_2, in the sense that the conclusion of a_1 is logically equivalent to the negation of some subset of the support of a_2. The formal definition is as follows.

$$\begin{array}{l} ATTACK_1(a_1, a_2) \leftrightarrow \\ \quad ARG_1(a_1) \wedge ARG_1(a_2) \wedge \\ \quad (\exists f \cdot FACT_1(supp(a_2), f) \wedge PRV_1(\emptyset, iff(conc(a_1), not(f))) \end{array}$$

Example 3. Suppose that Δ_0 is as defined in Example 1, above, and assume Δ_1 is constructed using the axioms and facts as above. Moreover, let $a_1 = \langle \ulcorner q \urcorner, \ulcorner p \rightarrow q \urcorner \rangle$ and let $a_2 = \langle \ulcorner \neg p \urcorner, \ulcorner t \rightarrow \neg p \urcorner \rangle$. Then $\Delta_1 \models_{\mathcal{L}_1} ATTACK_1(a_2, a_1)$.

Now, it is well-known that an attack relation is not in itself generally sufficient to resolve the issue of which arguments should be judged acceptable. Considering the various notions of acceptability from [7], for example, preferred extensions and grounded extensions always exist but may be empty, while stable extensions are never empty, but may not exist. More generally, however, Bench-Capon has argued, taking his cue from Perelman [14], that a logical approach is just too simplistic in many scenarios [1, pp.429–430]: to resolve the argument system, we need to consider the values that arguments appeal to, and make our judgements not only the logical soundness of the arguments, but also on how we rank the values embodied in arguments:

> Often, no conclusive demonstration of the rightness of one side is possible: both sides will plead their case, presenting arguments for their view as to what is correct. Their arguments may all be [logically] sound. But their arguments will not have equal value for the judge charged with deciding the case: the case will be decided by the judge preferring one argument over another. [...] One way of [justifying such preferences] is to relate the arguments to the purposes of the law under consideration, or the values that are promoted by deciding for one side against the other.

Bench-Capon goes on to show how Dung's argument framework may be extended with values, intended to capture such a system of arguments: we will proceed to formalise Bench-Capon's framework within $\mathcal{L}_1$. First, we assume that the domain of $\mathcal{L}_1$ contains a set of *values*. We shall not be concerned with the nature of such values, but examples might include, (taking from a legal setting), the right to life, the right to free speech, public interest, and the right to own property. Now, we will associate each argument with such a value, by means of a two-place $\mathcal{L}_1$ predicate $VAL_1(\cdots)$. We require that every possible $\mathcal{L}_1$ argument has a value.

$$ARG_1(a) \rightarrow \exists v \cdot VAL_1(a, v)$$

While one could in principle consider arguments being associated with more than one value, for simplicity we will assume that arguments have exactly *one* value.

$$ARG_1(a) \wedge VAL_1(a, v_1) \wedge VAL_1(a, v_2) \rightarrow (v_1 = v_2)$$

Next, we introduce *audiences*. We assume the domain of $\mathcal{L}_1$ contains a set of audiences: an audience, in Perelman and Bench-Capon's frameworks, is a group of agents who have *preferences* over values. We denote audiences by q, q', and so on, and use a ternary $\mathcal{L}_1$ predicate $v_1 \succ_q v_2$, with the intended meaning that audience q ranks value v_1 above value v_2. The $\succ_a$ relation is assumed to be transitive, irreflexive, and asymmetric, giving the following three axioms for Δ_1.

$$\begin{array}{l} ((v_1 \succ_q v_2) \wedge (v_2 \succ_q v_3)) \rightarrow (v_1 \succ_q v_3) \\ \neg(v_1 \succ_q v_1) \\ (v_1 \succ_q v_2) \rightarrow \neg(v_2 \succ_q v_1) \end{array}$$

We have a ternary $DEFEATS_1(\cdots)$ predicate, with the idea being that $DEFEATS_1(a_1, a_2, q)$ if argument a_1 attacks a_2 and it is not the case that the value promoted by a_1 is ranked over that promoted by a_2 for audience q.

$$\begin{aligned}&\mathit{DEFEATS}_1(a_1, a_2, q) \leftrightarrow\\&\quad \mathit{ATTACK}_1(a_1, a_2) \wedge\\&\quad \mathit{VAL}_1(a_1, v_1) \wedge \mathit{VAL}(a_2, v_2) \rightarrow \neg(v_2 \succ_q v_1)\end{aligned}$$

We will assume that some appropriate axiomatization is given in $\mathcal{L}_1$ for working with sets of arguments, defining set membership for arguments ("$a \in A$") and subsets ("$A_1 \subseteq A_2$") – the axiomatization is standard, and we thus omit it. We then say an argument a is acceptable with respect to a set of arguments A for audience q if every possible argument that defeats q is itself defeated for q by some member of A [1]. We characterise this via the $\mathcal{L}_1$ predicate $\mathit{ACCEPTABLE}_1(\cdots)$.

$$\begin{aligned}&\mathit{ACCEPTABLE}_1(a_1, A, q) \leftrightarrow\\&\quad \forall a_2 \cdot \mathit{DEFEATS}_1(a_2, a_1, q) \rightarrow\\&\qquad \exists a_3 \cdot (a_3 \in A) \wedge \mathit{DEFEATS}_1(a_3, a_2, q)\end{aligned}$$

A set of arguments A is conflict free for audience q if for every pair of arguments a_1, a_2, either it is not the case that a_1 defeats a_2, or else a_2 is ranked over a_1 by q.

$$\begin{aligned}&\mathit{CFREE}_1(A, q) \leftrightarrow\\&\quad (a_1 \in A) \wedge (a_2 \in A) \rightarrow\\&\qquad ((\neg \mathit{DEFEATS}_1(a_1, a_2, q)) \vee\\&\qquad (\mathit{VAL}_1(a_1, v_1) \wedge \mathit{VAL}_1(a_2, v_2) \rightarrow (a_2 \succ_q a_1)))\end{aligned}$$

A set of arguments A that is conflict free for audience q is *admissible* if every argument in the set if acceptable with respect to A.

$$\begin{aligned}&\mathit{ADM}_1(A, q) \leftrightarrow\\&\quad \mathit{CFREE}_1(A, q) \wedge\\&\quad \forall a \cdot (a \in A) \rightarrow \mathit{ACCEPTABLE}_1(a, A, q)\end{aligned}$$

Finally, a set of arguments A is a *preferred extension* for audience q if it is a maximal (with respect to set inclusion) admissible set with respect to q.

$$\begin{aligned}&\mathit{PE}_1(A_1, q) \leftrightarrow\\&\quad \mathit{ADM}_1(A_1, q) \wedge \forall A_2 \cdot (A_1 \subseteq A_2) \rightarrow \neg \mathit{ADM}_1(A_2, q)\end{aligned}$$

Thus far, we have shown how a logic based argument system can be developed within our framework that combines such frameworks with Dung's and Bench-Capon's systems. Note that in order to do this, we have frequently defined predicates that take as their argument formulae and sets of formulae: and any mathematically sound framework which achieved this would inherently have to be meta-logical.

3.3 Meta-Arguments: Δ_2

To construct Δ_2, we proceed much as we did when constructing Δ_1. (Note that in this section, many of the definitions are exact analogues of those appearing at level 1, with the predicate subscripts simply changed from 1 to 2: we will omit such definitions when there is no possibility of ambiguity.) First, we will have a constant Δ_1, which will denote the level 1 theory constructed as above (and of course, this level 1 theory was constructed with respect to the level 0 theory Δ_0, containing object-level sentences). We use a quoting convention for formulae in exactly the same way that we used such a convention in Δ_1, and introduce predicates $FACT_2(\cdots)$ and $PRV_2(\cdots)$ and subset relation $\subseteq$ as above. Also analogously to Δ_1, we assert that every statement appearing in Δ_1 is a $FACT_2(\cdots)$ of Δ_1:

$$FACT_2(\Delta_1, f) \quad \text{for each } f \in \Delta_1$$

We also construct a predicate $ARG_2(\cdots)$, which characterises an argument at level 2 of the hierarchy, by way of predicates $PF_2(\cdots)$ (for prima facie level 2 arguments), and $CPF_2(\cdots)$ (for consistent prima facie level 2 arguments), again following the pattern established at level 1.

Example 4. Suppose that Δ_0 is as defined in Example 1, above, and Δ_1 and Δ_2 are constructed as indicated above. Then we can conclude the following.

$$\begin{aligned}
&\Delta_2 \models_{\mathcal{L}_2} \\
&\quad \exists T.(T \subseteq \Delta_1) \wedge \\
&\quad ARG_2(\langle \ulcorner ARG_1(\langle \ulcorner q \urcorner, \{\ulcorner p \urcorner, \ulcorner p \rightarrow q \urcorner\}\rangle)\urcorner, T\rangle)
\end{aligned}$$

$$\begin{aligned}
&\Delta_2 \models_{\mathcal{L}_2} \\
&\quad \exists T.(T \subseteq \Delta_1) \wedge \\
&\quad ARG_2(\langle \ulcorner ARG_1(\langle \ulcorner \neg p \urcorner, \{\ulcorner t \urcorner, \ulcorner t \rightarrow \neg p \urcorner\}\rangle)\urcorner, T\rangle)
\end{aligned}$$

$$\begin{aligned}
&\Delta_2 \models_{\mathcal{L}_2} \\
&\quad \exists T.(T \subseteq \Delta_1) \wedge \\
&\quad ARG_2(\langle \ulcorner ARG_1(\langle \ulcorner \neg s \urcorner, \{\ulcorner p \urcorner, \ulcorner p \rightarrow q \urcorner, \ulcorner (q \vee r) \rightarrow \neg s \urcorner\}\rangle)\urcorner, T\rangle)
\end{aligned}$$

This example may at first sight not appear to be saying anything more interesting than was said at level 1: indeed, it looks rather like we are saying, in a fancy way, that certain structures may be proved to be level 1 arguments – which we could also say at level 1! To illustrate the value of this construction, let us therefore take apart the reasoning process through which we can assert that a structure is an argument in Δ_2.

Suppose that, for some $\mathcal{L}_0$ formula f and set of $\mathcal{L}_0$ formulae T_1, we have the following:

$$\Delta_2 \models_{\mathcal{L}_2} ARG_2(\langle \ulcorner ARG_1(\langle f, T_1\rangle)\urcorner, T_2\rangle)$$

This is stating that we can prove that in level 2 that $ARG_1(\langle f, T_1\rangle)$: T_1 serves as the support of this argument, and will be a minimal set of $\mathcal{L}_0$ formulae from the domain theory sufficient to establish the $\mathcal{L}_0$ conclusion f.

But what exactly is T_2 here? It is not a set of $\mathcal{L}_0$ formulae, because we are working in Δ_2. T_2 serves as the support for the conclusion $ARG_1(\langle f, T_1\rangle)$: as the subscript indicates, this conclusion is a sentence of $\mathcal{L}_1$, and so the support is a set of $\mathcal{L}_1$ sentences. What will this support look like? That is, what will T_2 contain? It will contain a minimal consistent set of $\mathcal{L}_1$ sentences that are sufficient to establish the conclusion $ARG_1(\langle f, T_1\rangle)$. In particular, T_2 *must contain* *a minimal set of sentences from Δ_1 that are required to prove that the structure is an argument: in particular, the $\mathcal{L}_1$ axioms corresponding to proof rules that are required to establish this conclusion, and the axioms corresponding to the definition of an argument.*

But of course, this lays bare the mechanism by which we can establish a statement such as $ARG_1(\langle f, T_1\rangle)$: when we are presented with an argument at $\mathcal{L}_2$ to the effect that something is an $\mathcal{L}_1$ argument, we can examine the support to see *how* this conclusion is justified. This justifies our claim, above, that at level 2, we can not only state that a particular structure is an argument, but also we can characterise *the means by which we can assert this*, i.e., the mechanism of establishing that something is an argument. This is critical if we want to consider the axioms and rules that were used to construct the argument.

To see the value of this, let us consider, for example, an *intuitionistic* argument to be one that can be established without the use of the axiom $\neg\neg f \rightarrow f$ (cf. [6]). Recall that in our Δ_1 axiomatization, we included an axiom capturing this axiom, which is valid in classical logic, but is not valid in intuitionistic logic. Let RA be an $\mathcal{L}_2$ constant that denotes this $\mathcal{L}_1$ axiom. We can define an $\mathcal{L}_2$ predicate $IARG_2(\cdots)$ which characterises an $\mathcal{L}_2$ argument that can be constructed *without RA*.

$$\begin{aligned}&IARG_2(a) \leftrightarrow\\&\quad ARG_2(a) \wedge \neg(FACT_2(supp(a), RA))\end{aligned}$$

In general, there will of course be cases where we have $ARG_2(\langle\ulcorner ARG_1(a)\urcorner, T\rangle)$ but not $IARG_2(\langle\ulcorner ARG_1(a)\urcorner, T\rangle)$.

In the same way, we can define conclusions that can *only* be established by means of classical constructions, i.e., cannot be established intuitionistically. Let us define a unary $\mathcal{L}_2$ predicate $SCARG_2(\ldots)$, which takes an $\mathcal{L}_1$ argument, and which is true when this argument can *only* be established classically.

$$\begin{aligned}&SCARG_2(a) \leftrightarrow\\&\quad \forall T \cdot ((T \subseteq \Delta_1) \wedge ARG_2(\langle\ulcorner ARG_1(a)\urcorner, T\rangle)) \rightarrow\\&\qquad \neg IARG_2(\langle\ulcorner ARG_1(a)\urcorner, T\rangle)\end{aligned}$$

Again, we note that this type of construction cannot be achieved at lower levels of the hierarchy.

4 Conclusions

We have argued that the any proper formal treatment of logic-based argumentation must be a *meta-logical* system. This is because formal arguments and dialogues do not just involve asserting the truth or falsity of statements about some domain of discourse: they involve making arguments *about* arguments, and potentially higher-level references (i.e., arguments about arguments about arguments). To illustrate this meta-logic approach to argumentation, and provide a proof of concept for it, we developed a formalisation of argumentation using a hierarchical first-order meta-logic. We defined three tiers of a hierarchical argument system, with the level 0 of this hierarchy corresponding to object-level statements about the domain, level 1 defining the notion of an argument, and capturing notions of attack/defeat, values and audiences, and the acceptability of argument sets. At level 2 of the hierarchy, we are able to reason about the process of asserting that a particular structure represents an argument, and how such an assertion is constructed. In particular, we are able to capture at level 2 the axioms/rules that must be used in order to construct an argument, and hence distinguish between arguments constructed in different ways.

Although meta-*logical* systems have been widely studied in the past four decades, comparatively little research appears to have addressed the issue of meta-*argument*. One notable exception is the work of Brewka, who in his [4], presented a tiered argument system which at first sight appears to have much in common with our own. However, although there are several points of similarity, there are also many differences, and the motivation and ultimate formalisation in Brewka's approach is in fact rather different.

There are several potentially interesting avenues for future work. First, it we believe it would be straightforward to implement such a hierarchical argument system: in particular, PROLOG has been found to be an extremely useful tool for meta-logical reasoning and the implementation of meta-interpreters for logics [21]. Second, our system currently has no notion of dialogue or argumentation protocol: again, it would be straightforward to extend the framework with dialogues, axiomatizing the protocol rules within the system. Third, it would be useful to extend the framework to include reasoning about each agent's beliefs and intentions, as in [12]: as demonstrated in [10, 9], (hierarchical) meta-logic can be an extremely useful tool for this purpose.

Acknowledgments

This work was made possible by funding from NSF #REC-02-19347, NSF #IIS 0329037, and the EC's IST programme under the "ASPIC" project. We gratefully acknowledge the comments of ASPIC researchers Trevor Bench-Capon and Sylvie Doutre, as well as the anonymous reviewers, which have helped us to improve this paper significantly.

References

1. T. J. M. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
2. P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.
3. A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1-2):63–101, 1997.
4. G. Brewka. Dynamic argument systems: A formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation*, 11(2): 257–282, 2001.

5. S. Costantini. Meta-reasoning: A survey. In A. C. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond – Essays in Honour of Robert A. Kowalski (LNAI Volumes 2408)*, pages 253–288. Springer-Verlag: Berlin, Germany, 2002.
6. M. Dummett. *Elements of Intuitionism.* Oxford University Press: Oxford, England, 1977.
7. 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.
8. M. R. Genesereth and N. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann Publishers: San Mateo, CA, 1987.
9. J. Grant, S. Kraus, and D. Perlis. A logic for characterizing multiple bounded agents. *Autonomous Agents and Multi-Agent Systems*, 3(4):351–387, 2000.
10. K. Konolige. A first-order formalization of knowledge and action for a multiagent planning system. In J. E. Hayes, D. Michie, and Y. Pao, editors, *Machine Intelligence 10*, pages 41–72. Ellis Horwood: Chichester, England, 1982.
11. P. Krause, S. Ambler, M. Elvang-Gøransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995.
12. S. Parsons, C. A. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
13. S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.
14. C. Perelman and L. Olbrechts-Tyteca. *The New Rhetoric: A Treatise on Argumentation.* University of Notre Dame Press: Notre Dame, 1969.
15. D. Perlis. Languages with self reference I: Foundations. *Artificial Intelligence*, 25:301–322, 1985.
16. D. Perlis. Meta in logic. In P. Maes and D. Nardi, editors, *Meta-Level Architectures and Reflection*, pages 37–49. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1988.
17. J. L. Pollock. Justification and defeat. *Artificial Intelligence*, 67:377–407, 1994.
18. H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic (second edition).* Kluwer Academic Publishers: Dordrecht, The Netherlands, 2001.
19. C. Reed. Dialogue frames in agent communication. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, pages 246–253, Paris, France, 1998.
20. C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In M. P. Singh, A. Rao, and M. J. Wooldridge, editors, *Intelligent Agents IV (LNAI Volume 1365)*, pages 177–192. Springer-Verlag: Berlin, Germany, 1998.
21. L. Sterling and E. Shapiro. *The Art of Prolog (Second Edition).* The MIT Press: Cambridge, MA, 1994.
22. S. Toulmin. *The Uses of Argument.* Cambridge University Press: Cambridge, England, 1958.
23. R. Turner. *Truth and Modality for Knowledge Representation.* Pitman Publishing: London, 1990.
24. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* State University of New York Press, Albany, NY, 1995.

Nested Argumentation and Its Application to Decision Making over Actions

S. Modgil

Advanced Computation Lab, Cancer Research UK, London WC2A 3PX
sm@acl.icnet.uk

Abstract. In this paper we describe a framework in which the grounds for one argument's defeat of another is itself subject to argumentation. Hence, given two conflicting arguments, each of which defeat the other, one can then determine the preferred defeat and hence the preferred argument. We then apply this nested argumentation to selection of an agent's preferred 'instrumental' arguments, where each such argument represents a plan of actions for realising an agent's goals.

1 Introduction

There is a growing body of work addressing the uses of argumentation in agent applications. Many of these works define an argumentation system for construction of arguments, and then instantiate Dung's framework [6] to determine which arguments are 'justified' or 'preferred' on the basis of the ways in which they interact. The interactions considered include the binary relations of *attack* and *defeat*. The former represents that two arguments conflict with each other. The latter additionally accounts for some relative valuation of the strength of two attacking arguments. However, given two mutually attacking arguments *A1* and *A2*, it may well be that there are grounds for *defeat(A1,A2)* **and** *defeat(A2,A1)*. For example, strengths of arguments may be evaluated on the basis of different criteria, so that *A1* defeats *A2* based on criterion c, and *A2* defeats *A1* based on criterion c'. Also, for any given criterion, evaluation of an argument's strength may vary according to the context in, or the perspective from, which it is evaluated. For example, reference to one information source for determining argument strength may indicate that *A1* defeats *A2*, whereas from the perspective of another information source, *A2* may defeat *A1*. Given two 'conflicting defeats' *defeat(A1,A2)* and *defeat(A2,A1)*, then one cannot establish which of *A1* or *A2* is preferred. However, such a preference can be established if one can determine which *defeat is preferred*.

We therefore propose that the reasoning underlying relative evaluation of the strength of two attacking arguments should itself be subject to argumentation. Hence, one constructs two 'level 2' arguments *B1* and *B2*, respectively providing grounds for *defeat(A1,A2)* and *defeat(A2,A1)*. To determine which of these conflicting defeats is preferred, we need to determine a preference between the mutually attacking arguments *B1* and *B2*. This in turn requires construction of 'level 3' arguments: *C1* providing grounds for *defeat(B1,B2)* or *C2* providing

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 57–73, 2006.

grounds for *defeat(B2,B1)*. Of course, one might be able to construct both *C1* and *C2*, in which case one ascends to another level to determine which of these are preferred. In principle, this nested argumentation can continue indefinitely. Reasoning about the relative strength of arguments is also explored in [9, 11]. They do so by extending the object level language for argument construction with rules that allow context dependent inference of possibly conflicting relative prioritisations of rules. Thus, argument strength is exclusively based on rule priorities. The framework proposed here allows for argument strength to be based on any number of criteria. Furthermore, our framework formalises reasoning about the strength and defeats amongst arguments at the meta rather than object level. These requirements are of particular relevance to the use of argumentation in agent applications.

The issue of conflicting defeats is particularly relevant for agent applications, given the general requirement for a context dependent account of agents' cognitive processes. Specifically, a number of recent works [1, 2, 4, 8, 9] extend theories of argumentation over beliefs, to argumentation over agents' desires and intentions. For example, Amgoud [1, 2], and subsequently Hulstijn [8], define construction of *instrumental* arguments composed of actions and sub-goals for realising some top level goal (these arguments can be thought of as unscheduled plans). The idea is to then choose the preferred instrumental arguments so as to determine which plans the agent should adopt. However, the argumentation systems proposed do not straightforwardly instantiate Dung's framework. Furthermore, given conflict free sets of instrumental arguments, the preferred sets are chosen solely on the basis of those that maximise the number of agent goals realised. However, in practical settings, strengths of arguments need to be established on the basis of multiple additional criteria such as the efficacy and temporal and financial costs of a plan's actions with respect to their goals. This implies a need to handle conflicting defeats in order to determine the preferred instrumental arguments. This need may also be a requirement for argumentation-based multi-agent dialogues [12], where the agents represent different perspectives from which communicated arguments are evaluated.

The main contributions of this paper are as follows. In section 2 we formalise nested argumentation over nested Dung argumentation frameworks. In section 3 we modify and build on Amgoud's system [1, 2] for constructing instrumental arguments. In particular our system is able to instantiate a Dung framework without adapting Dung's central definitions. In section 4 we apply nested argumentation to decide the preferred instrumental arguments on the basis of multiple information sources and criteria. In section 5 we conclude with a discussion of related and future work.

2 Nested Argumentation

Arguments can be said to *rebut* attack or *undercut* attack. In the former case the attack is symmetric; *attack(A1,A2)* and *attack(A2,A1)*. An example of a rebut attack is when the claim of *A1* conflicts with the claim of *A2*. Defeat additionally accounts for some relative valuation of the strength of attacking arguments: *defeat(A1,A2)* if *attack(A1,A2)* and it is not the case that *A2* is stronger than *A1*. Hence, in the case of a rebut attack, *defeat(A1,A2)* **and** *defeat(A2,A1)* if: **i)** their are no grounds for determining the relative strengths of *A1* and *A2*, or **ii)** there are grounds for *A1* being stronger than *A2*, **and** grounds for *A2* being stronger than *A1*.

Unlike rebut attacks, undercut attacks are asymmetric; *attack(A1,A2)* but not *attack(A2,A1)*. We support the view ([3, 11]) that one should not distinguish between undercut attacks and defeats; i.e., undercut defeats should not depend on the relative strength of arguments. To illustrate, consider a Pollock undercut defeat [10] whereby the claim of argument *A1* denies that the premises of *A2* support its claim (an attack on the link between premises and claim of *A2*). Pollock requires that *A2* is not stronger than *A1*. This leads to unintuitive results: if *A2* is stronger than *A1*, or information regarding their relative strength is missing, then neither argument defeats or attacks each other, and hence both arguments can be coherently held to be acceptable.

As discussed in section 1, we aim at a framework in which argumentation over the grounds for one argument being stronger than another can be used to resolve conflicting defeats of type **ii)** above. In this way one can determine a preference amongst mutually defeating arguments. We begin with two notions of a Dung argumentation framework, and then give Dung's standard definition of the preferred extensions of an argumentation framework.

Definition 1. *Let Args be a finite set of arguments. An argumentation framework AF is a pair (Args, Attack), where Attack* $\subseteq$ *(Args* $\times$ *Args). A justified argumentation framework JAF is a pair (Args, Defeat), where Defeat* $\subseteq$ *(Args* $\times$ *Args).*

Definition 2. *For any set of arguments S:*

- *S is **conflict free** iff no argument in S is defeated(attacked) by an argument in S.*
- *An argument A is **acceptable** w.r.t. S iff each argument defeating (attacking) A is defeated (attacked) by an argument in S.*
- *A conflict free set of arguments S is **admissible** iff each argument in S is acceptable with respect to S.*
- *A conflict free set of arguments S is a **preferred extension** iff it is a maximal (w.r.t. set inclusion) admissible set.*

Definition 3. *Let* $\{S_1, \ldots, S_n\}$ *be the preferred extensions of JAF = (Args,Defeat)*[1]*. Then* $\bigcap_{i=1}^{n} S_i$ *is the set of preferred arguments of JAF (denoted **Pf**(JAF))*

[1] Note that there will be a finite number of preferred extensions given the restriction in definition 1 to argumentation frameworks with a finite number of arguments.

We now define nested argumentation frameworks of the form $(AF_1, \ldots, AF_n)$. We make some minimal assumptions about the argumentation system instantiating each AF. In particular, each argument A in a system has a claim *claim(A)* (we write *claims(S)* to denote $\{claim(A) \mid A \in S\}$), and for AF_i, $i > 1$, the language for argument construction is a first order language whose signature contains the binary predicate symbol *defeat* and a set of constants $\mathsf{f\_name}_{i-1}(Args_{i-1}) = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ naming arguments in $Args_{i-1}$.

Definition 4. *A nested argumentation framework (NAF) is an ordered finite set of argumentation frameworks ((Args*$_1$*, Attack*$_1$*),. . .,(Args*$_n$*, Attack*$_n$*)) such that for i = 1 . . . n-1,* $Attack_i \supseteq \{(A,A') \mid defeat(\mathcal{A}, \mathcal{A}') \in claims(Arg_{i+1})\}$.

Given a NAF $(AF_1,\ldots,AF_n)$, we now define a justified NAF, mapping each AF_i to a JAF_i. Intuitively, an AF_{i+1} argument B with claim $defeat(\mathcal{A}', \mathcal{A})$ provides the grounds for an AF_i argument A' being stronger than A. The basic idea is that an attack (A,A') in some AF_i is not a defeat in JAF_i iff an argument B with claim $defeat(\mathcal{A}', \mathcal{A})$ is a preferred argument of JAF_{i+1}.

Definition 5. *Let* $\triangle = (AF_1, \ldots, AF_n)$ *be a NAF. Then the justified NAF* $(JAF_1, \ldots, JAF_n)$ *is defined as follows:*
1) *For i = 1. . .n,* $Args_i$ *in* JAF_i $=$ $Args_i$ *in* AF_i
2) $Defeat_n = Attack_n$
3) *For i = 1. . .n-1,* $Defeat_i = Attack_i - \{(A,A') \mid defeat(\mathcal{A}', \mathcal{A}) \in claims(\boldsymbol{Pf}(JAF_{i+1}))\}$
We say that $\boldsymbol{Pf}(JAF_1)$ *is the set of preferred arguments of* $\triangle$.

Note that the restriction in definition 4 ensures that any undercut attack in AF_i will, as required, be an undercut defeat in JAF_i:

Proposition 1. *Let* $(JAF_1, \ldots, JAF_n)$ *be defined on the basis of* $(AF_1, \ldots, AF_n)$. *Then, for* $i = 1 \ldots n$: $(A, A') \in Attack_i$ *and* $(A', A) \notin Attack_i$ *implies* $(A, A') \in Defeat_i$ *and* $(A', A) \notin Defeat_i$.

Proof. Suppose otherwise: i.e., $(A, A') \notin Defeat_i$ *or* $(A', A) \in Defeat_i$. *If* $(A, A') \notin Defeat_i$, *then by def.5(3),* $defeat(\mathcal{A}', \mathcal{A}) \in claims(\boldsymbol{Pf}(JAF_{i+1}))$. *By def.5(1) the arguments in* AF_{i+1} *are the same as those in* JAF_{i+1}. *Hence,* $defeat(\mathcal{A}', \mathcal{A})$ *is the claim of an argument in* AF_{i+1}. *Hence,* $(A', A) \in Attack_i$ *by the restriction* $Attack_i \supseteq \{(A',A) \mid defeat(\mathcal{A}', \mathcal{A}) \in claims(Arg_{i+1})\}$ *in def.4. This contradicts the assumption that* $(A', A) \notin Attack_i$. *If* $(A', A) \in Defeat_i$, *then by def.5(3)* $(A', A) \in Attack_i$, *again contradicting the assumption that* $(A', A) \notin Attack_i$.

Proposition 2. *Let* $(JAF_1, \ldots, JAF_n)$ *be defined on the basis of* $(AF_1, \ldots, AF_n)$. *Assuming* $defeat(\mathcal{A}',\mathcal{A}) \in claims(\boldsymbol{Pf}(JAF))$ *implies* $defeat(\mathcal{A}, \mathcal{A}') \notin claims\ (\boldsymbol{Pf}(JAF))$ *(since arguments for these claims conflict and so cannot both be in the preferred set), then for* $i = 1 \ldots n$:

E is a conflict free maximal subset of Args in AF_i *iff E is a conflict free maximal subset of* $Args'$ *in* JAF_i.

Proof. *By def.5 $Args = Args'$. It remains to show that: A* ***attacks, or is attacked by, an argument in*** *AF_i* ***iff*** *A* ***defeats, or is defeated by, an argument in*** *JAF_i.*

For $i = n$ this follows from def.5(2). For $i \neq n$, the right to left half follows from def.5(3) which implies that $Defeat_i \subseteq Attack_i$. For the left to right half, consider two cases: i) $(A, A') \in Attack_i$, $(A', A) \notin Attack_i$; ii) $(A, A') \in Attack_i$, $(A', A) \in Attack_i$. Case i) is given by proposition 1. For case ii), we show that (A, A') or $(A', A) \in Defeat_i$. Suppose otherwise. Then by def.5(3), $defeat(\mathcal{A}', \mathcal{A})$ and $defeat(\mathcal{A}', \mathcal{A}) \in claims(\boldsymbol{Pf}(JAF_{i+1}))$, contradicting the assumption.

Given proposition 2, the preferred extensions of JAF_i will be a subset of those of AF_i. It is nested argumentation's substitution of rebut attacks in AF_i by asymmetric defeats in JAF_i that enables choice of a single preferred extension. In the following examples we write $A1 \rightleftharpoons A2$ to denote rebut attacks *attack*(*A1*,*A2*) and *attack*(*A2*,*A1*), and $A1 \rightharpoonup A2$ for the asymmetric undercut *attack*(*A1*,*A2*).

Example 1. Let $\triangle = (AF1, AF2, AF3)$ where:
$AF1 = (\{A1, A2, A3, A4, A5\}, \{A1 \rightleftharpoons A2, A2 \rightharpoonup A3, A4 \rightleftharpoons A5\})$,
$AF2 = (\{B1, B2, B3, B4\}, \{B1 \rightleftharpoons B2, B4 \rightharpoonup B3\})$, where *claim(B1)* = *defeat*($\mathcal{A}$1,$\mathcal{A}$2), *claim(B2)* = *defeat*($\mathcal{A}$2,$\mathcal{A}$1), *claim(B3)* = *defeat*($\mathcal{A}$4,$\mathcal{A}$5)
$AF3 = (\{C1\}, \emptyset)$ where *claim(C1)* = *defeat*($\mathcal{B}$1,$\mathcal{B}$2).

Then: $\boldsymbol{Pf}(JAF_3) = \{C1\}$, $\boldsymbol{Pf}(JAF_2) = \{B1, B4\}$, $\boldsymbol{Pf}(JAF_1) = \{A1, A3\}$ - the set of preferred arguments of $\triangle$. Notice that *B4*'s undercut of *B3* means that $A4$ is not preferred, despite the fact that there exists no *AF2* argument for *defeat*($\mathcal{A}$5,$\mathcal{A}$4). If *B3* were not undercut then $A4$ would also be preferred.

We consider the above to be a general framework for modelling nested argumentation, whereby given a particular argumentation system instantiating *AF1*, one can define suitable mappings from AF_i to AF_{i+1}, and logics for construction of arguments instantiating AF_i, $i > 1$. In what follows we show how this is possible, applying nested argumentation to decision making over plans of action.

3 A System for Constructing Instrumental Arguments

In [1, 2], Amgoud describes how realisation trees for an agent's initial goals can be built from an agent's planning rules. These rules are of a single type, relating goals to their sub-goals, and (sub)goals to the actions they are realised by. These realisation trees are modelled as 'instrumental' arguments for a claim - the initial goal - where the supporting argumentation can be thought of as a plan of actions and subgoals for realising the initial goal. Argument theoretic notions are then used to select the preferred arguments from a set of arguments that may conflict given constraints precluding joint execution of plans. Here we define a modified system for construction of instrumental arguments.

In what follows we define an agent description consisting of formulae in some propositional language $\mathcal{L}1$, where, unlike [1, 2], we distinguish three types of planning rule, and distinguish between literals denoting beliefs, atomic actions (that need no further plan to be achieved) and goals that require further plans to be achieved:

Definition 6. *Let $\mathcal{L}1$ be a propositional language consisting of three sets Ac, G and B of propositional literals denoting actions, goals and beliefs respectively. Let $\bigwedge \overline{Ac}$ ($\bigwedge \overline{G}$) ($\bigwedge \overline{B}$) denote the conjunction of a (possibly empty) subset of literals in Ac (G) (B). A planning rule is of the form $r : (l_1 \wedge \ldots \wedge l_{n-1}) \Rightarrow l_n$, where r is a unique propositional name for the rule, and for $i = 1 \ldots n$, l_i is a propositional literal or its negation. We write head(r) to denote $\{l_n\}$ and body(r) to denote $\{l_1, \ldots, l_{n-1}\}$. There are three types of planning rule:*

1. *precondition-action rules - $\bigwedge \overline{B} \Rightarrow l_n$ where $l_n \in Ac$*
2. *action-effect rules - $(\bigwedge \overline{B}) \wedge (\bigwedge \overline{Ac}) \Rightarrow l_n$ where $l_n \in B$ and $\overline{Ac}$ is non-empty*
3. *goal-realisation rules - $(\bigwedge \overline{B}) \wedge (\bigwedge \overline{Ac}) \wedge (\bigwedge \overline{G}) \Rightarrow l_n$ where $l_n \in G$*

Definition 7. *Let $\langle IG, \mathcal{B}, \mathcal{B}_p \rangle$ denote an agent description, where IG is the agent's set of initial goals ($IG \subseteq G$), the belief base $\mathcal{B}$ is a set of wff of $\mathcal{L}1$, and $\mathcal{B}_p$ is a set of planning rules.*

Note that planning rules are not material implications but behave as production rules. Intuitively, the antecedent $\bigwedge \overline{B}$ of a precondition-action rule represents what must be believed true about the current state of the world for an action to be applicable (i.e., the actions's preconditions). For action-effect rules, $\bigwedge \overline{B}$ represents what must be believed true about the world for actions $\bigwedge \overline{Ac}$ to result in some belief *b* to be true (i.e., *b* represents a postcondition or immediate effect of an action or actions). Finally, a goal-realisation rule represents that the goal in the head of the rule is realisable if the beliefs (effects of actions) in the antecedent are true and/or actions in the antecedent are executed and/or subgoals in the antecedent are realised.

Example 2. Let $\triangle$ be a medical agent description consisting of an initial treatment goal *g* and the planning rules: ***r1*** : *b1* ⇒ *a1*; ***r2*** : *a1* ⇒ *e1*; ***r3*** : *b2* ⇒ *a2*; ***r4*** : *a2* ⇒ *e1*; ***r5*** : *e1* ⇒ *g*, where *b1* (*b2*) represents a precondition for a medical action *a1* (*a2*), and *a1* (*a2*) results in an effect *e1* that realises *g*. For example, *a1* = 'administer aspirin', *a2* = 'administer chlopidogrel', *e1* is the effect 'reduced platelet adhesion' and *g* = 'prevent blood clotting'.

We now define a realisation tree *R* for an initial goal ($\vdash$ denotes classical consequence in this and subsequent definitions), where *root*(*R*) denotes the root node of *R*, $child_1(n), \ldots, child_k(n)$ denote the child nodes $n1, \ldots, nk$ of node *n*, and *n* is a leaf node if it has no child nodes. Also, a node *n* in *R* is the parent of a subtree *T* of *R* iff *child*(*n*) = *root*(*T*).

Definition 8. *A realisation tree based on $\langle IG, \mathcal{B}, \mathcal{B}_p \rangle$ is a finite AND tree R defined as follows:*

- *root(R) is a goal-realisation rule r where head(r)= g, $g \in IG$*
- *If node n of R is a planning rule $r : l_1 \wedge \ldots \wedge l_k \Rightarrow l$, then for i = 1...k :*

1. *if $l_i \in G$ or $l_i \in Ac$ then $child_i(n)$ is a planning rule r_i with head l_i*
2. *if $l_i \in B$, then if r is a precondition-action or action-effect rule, then $\mathcal{B} \vdash l_i$, else if r is a goal-realisation rule then $child_i(n)$ is an action-effect rule r_i with head l_i*

From hereon, *nodes(R)* returns the set of rules in R, *ig(R)* denotes the initial goal of R, and we refer to each node (rule) in R as a partial plan. Realisation trees as defined by Amgoud [1] and Hulstijn [8] are instrumental arguments. Two such arguments conflict, and so attack each other, if they contain partial plans that conflict.

Definition 9. *Two partial plans r_1 and r_2 conflict iff head(r_1) $\cup$ head(r_2) $\cup$ body(r_1) $\cup$ body(r_2) $\cup\ \mathcal{B} \cup \mathcal{B}_p \vdash \bot$.*

Hence, the defined arguments and their attacks can be used to instantiate a Dung framework. However, employing Dung's attack based definition of a conflict free set of arguments (def.2) may yield a preferred set of arguments that cannot be jointly adopted as plans. For example, suppose $\langle\ IG = \{a,b,c\}$, $\mathcal{B} = \{a' \wedge b' \rightarrow \neg c'\}$, $\mathcal{B}_p = \{a' \Rightarrow a,\ b' \Rightarrow b,\ c' \Rightarrow c\}\rangle$. Then the instrumental arguments as defined in [1, 8] are $R1 = (\Rightarrow a', a' \Rightarrow a,)$, $R2 = (\Rightarrow b', b' \Rightarrow b,)$, $R3 = (\Rightarrow c', c' \Rightarrow c,)$ (note that actions a',b',c' are not required to be the heads of planning rules in [1, 8]). No two arguments attack each other, and so the single preferred extension and hence set of preferred arguments is $\{R1, R2, R3\}$. However, the constraint in $\mathcal{B}$ precludes joint adoption of $R1$, $R2$ and $R3$.

This is rectified in Amgoud [2] by dropping the attack relation and attack based definition of conflict free sets. A conflict free set of instrumental arguments is simply defined on the basis that all the contained partial plans are mutually consistent. Thus, one obtains $\{R1, R2\}$, $\{R1, R3\}$, $\{R2, R3\}$. However, this represents a departure from Dung, so that in [2], the preferred extensions are selected solely on the basis of those sets that maximise the number of initial goals realised by the contained arguments (this is also the only criterion used in [1] and [8]). By this criterion, all the above sets are preferred extensions. Hence, none of the arguments are preferred.

The solution is to recognise that two or more realisation trees can be combined into a single instrumental argument provided that the trees do not conflict. We thus obtain instrumental arguments for more than one initial goal (conceptually, the conjunction of multiple initial goals can be considered as the head of a goal realisation rule whose body includes the individual initial goals). Thus, we will have three instrumental arguments $(R1 + R2)$, $(R1 + R3)$ and $(R2 + R3)$, each of which conflict with, and so attack, each other. We now define our notion of conflict free sets of realisation trees. Note that as in Hulstijn [8] (but unlike Amgoud), we additionally regard two realisation trees as conflicting if they realise the same goal. This is because an agent will at some stage have to decide and commit to a particular plan for realisation of any given goal.

Definition 10. *Let S be a set of realisation trees based on $\langle IG, \mathcal{B}, \mathcal{B}_p\rangle$. Then S is conflict free iff:*

- $\forall$ *R,R'* $\in$ *S, R* $\neq$ *R'* $\rightarrow$ *ig(R)* $\neq$ *ig(R')*
- $\bigcup_{R \in S} [\bigcup_{r \in nodes(R)} (head(r) \cup body(r))] \cup \mathcal{B} \cup \mathcal{B}_p \nvdash \perp$

An instrumental argument is defined as follows:

Definition 11. *Let S_1,...,S_m be the maximal (w.r.t set inclusion) conflict free sets of realisation trees based on $\langle IG, \mathcal{B}, \mathcal{B}_p \rangle$. Then $\{A_1, \ldots, A_m\}$ is the set of instrumental arguments based on $\langle IG, \mathcal{B}, \mathcal{B}_p \rangle$, where for $i = 1 \ldots m$, A_i is a finite AND tree with root node $n = \{ig(R) | R \in S_i\}$ and n is the parent of each tree in $\{R | R \in S_i\}$.*

Note that given definition of the planning rules (def.6) and realisation trees (def.8) one can readily show that:

Proposition 3. *Any path from the root to the leaf of an instrumental argument starts with the root node set of initial goals, followed by one or more goal-realisation rules, followed by at most one action-effect rule, and terminating in exactly one precondition-action rule.*

Each instrumental argument conflicts with and attacks all other instrumental arguments. We can now instantiate a Dung argumentation framework *AF1*:

Definition 12. *AF1 = (Args1, Attack1) where Args1 is the set of all instrumental arguments built from an agent description $\langle IG, \mathcal{B}, \mathcal{B}_p \rangle$, and Attack1 = $\{(A,A') | A, A' \in$ Args1 and $A \neq A'\}$.*

Example 3. In the following variation of an example in [2], an agent decides over plans of action to realise its initial goals to prepare for a journey to Africa (*pja*) and finish a paper (*fp*). Let the agent description be:

$\langle$ IG = $\{pja, fp\}$, $\mathcal{B} = \{w \rightarrow \neg\, pc\}$,
$\mathcal{B}_p = \{r1{:}w \Rightarrow fp,\ r2{:}t \wedge vac \Rightarrow pja,\ r3{:}int \Rightarrow t,\ r4{:}hop \Rightarrow vac,\ r5{:}pc \Rightarrow vac,$
$r6{:}dr \Rightarrow vac,\ r7{:} \Rightarrow int,\ r8{:} \Rightarrow dr,\ r9{:} \Rightarrow pc,\ r10{:} \Rightarrow hop,\ r11{:} \Rightarrow w\ \}\rangle$

where $G = \{fp, pja, t, vac\}$, $Ac = \{int, dr, pc, hop, w\}$, and w = 'work', pc = 'go to private clinic', t = 'get a ticket', vac = 'get vaccinated', dr = 'go to the doctor', hop = 'go to the hospital', int = 'log on to internet'. Note that

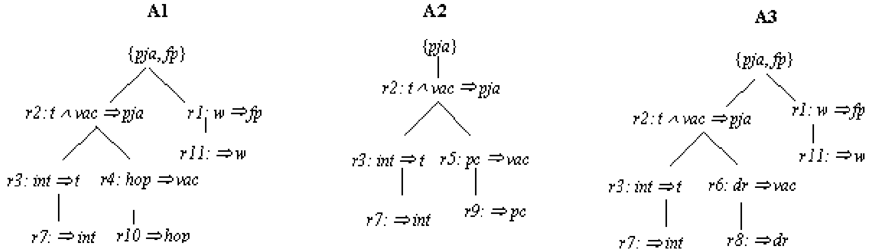


Fig. 1

$w \rightarrow \neg pc$ represents that working to finish the paper would take up to the end of the working day and so exclude going to a private clinic which (unlike the hospital and doctor's surgery) is closed outside of working hours.

Fig. 1 shows the arguments *Args1* based on $\langle IG, \mathcal{B}, \mathcal{B}_p \rangle$. $Attack1 = \{A1 \rightleftharpoons A2, A1 \rightleftharpoons A3, A2 \rightleftharpoons A3\}$. The preferred extensions of *AF1* = *(Args1,Attack1)* are: $\{A1\}$, $\{A2\}$, $\{A3\}$.

To summarise, an instrumental argument is a maximal conflict free set of realisation trees constructed from planning rules. Any two such arguments attack each other on the basis that they contain partial plans that conflict with each other and/or share an initial goal. This means that each maximal conflict free set of instrumental arguments (as defined by def.2) will always be a singleton set. We will have non-singleton sets when we consider other types of argument interacting with instrumental arguments. For example, arguments built from the agent's belief base may attack instrumental arguments by conflicting with beliefs in the antecedent of a precondition-action rule or action-effect rule.

Example 4. To illustrate, in our medical example 2, $AF1 = (\{A1,A2\},\{A1 \rightleftharpoons A2\})$ where *A1* is built from rules *r1*, *r2 r5*, and *A2* built from rules *r3*, *r4*, *r5*. An argument *A3* with claim $\neg b1$ would be a non-instrumental argument built from the agent's beliefs, which attacks *A1*. One might also account for the desirability of goals and effects realised or effected by an action. Assume the agent description is extended to include a set *U* of *undesirable* effects. Suppose an undesirable side-effect $e2 \in U$, and an action-effect rule $r6$: $b_1,\ldots,b_n,\ a1 \Rightarrow e2$, which represents that action *a1* has effect *e2* if $b_1, \ldots, b_n$ are believed true (e.g., aspirin has the effect gastric ulceration if it is believed that the patient has a history of gastritis). If $\mathcal{B} \vdash b_1, \ldots, b_n$ then *r6* will be used to construct a non-instrumental argument attacking *A1*.

However, the focus of this paper is on determining preferences amongst instrumental arguments that mutually attack and defeat each other, given that the strength of such arguments can be valuated on the basis of different criteria, or for any given criterion, on the basis of different sources. In the following section we show how nested argumentation can be used to resolve these conflicting defeats and thus determine a single preferred instrumental argument.

4 Applying Nested Argumentation to Decide the Preferred Instrumental Arguments

In what follows we define a *NAF* (AF_1,AF_2,AF_3) where AF_1 is defined as in the previous section. Arguments instantiating AF_2 will be for valuations of the strengths of AF_1 arguments and defeats between AF_1 arguments. Arguments instantiating AF_3 will make use of orderings on sources and criteria to construct arguments for defeats between AF_2 arguments. We then apply nested argumentation to determine a single preferred instrumental argument.

4.1 Defining the Argumentation Framework AF_2

Firstly, we define an argumentation system instantiating AF_2. We define the language $\mathcal{L}2$, a logic for argument construction, and a definition of conflict (attack).

Definition 13. *Let AF1 = (Args1, Attack1) be defined by an agent description $\langle IG, \mathcal{B}, \mathcal{B}_p \rangle$. Then $\mathcal{L}2$ is any first order logic language whose signature contains the set of real numbers $\Re$, the binary predicate symbols "attack" and "defeat", the arithmetic less than relation "$<$", and the following sets of constant symbols:*
- a set of argument names $\mathsf{f\_name}_1(Args1)$
- the set of planning rule names $\{r \mid r : l_1 \wedge \ldots \wedge l_k \Rightarrow l \in \mathcal{B}_p\}$
- a set Π denoting criteria and a set Ψ denoting sources

In what follows, variables $X, Y, \ldots$ range over $\Re$, $\mathcal{A}, \mathcal{A}_1, \mathcal{A}_2 \ldots$ range over $\mathsf{f\_name}_1(Args1)$, $P, P_1, P_2 \ldots$ range over criteria, $S, S_1, S_2 \ldots$ range over sources, and lower case roman letters range over all other constants in $\mathcal{L}2$. Lower case greek letters range over predicate formulae in $\mathcal{L}2$. Also, $\vdash_{FOL}$ denotes first order classical inference, and for any first order theory we assume the usual axiomatisation of $<$. We now define a mapping from *AF1* to a set $\triangle_{map}$ of first order implications and ground predicates in $\mathcal{L}2$. In this way an instrumental argument A is decomposed into its 'sub-arguments', e.g., the initial goals of A, or actions and action goal pairs in A.

Definition 14. *Let AF1 = (Args1, Attack1). Then $\triangle_{map}$ is defined as follows:*

- *attack($\mathcal{A}, \mathcal{A}'$) $\in \triangle_{map}$ iff (A,A$'$) $\in$ Attack1*
- *initial_goal($\mathcal{A}$,g) $\in \triangle_{map}$ iff A $\in$ Args1, g $\in$ root(A)*
- *goal($\mathcal{A}$,g) $\in \triangle_{map}$ iff A $\in$ Args1, r: $l_1 \wedge \ldots \wedge l_k \Rightarrow g$ is a node in A and g $\in$ G*
- *action($\mathcal{A}$, a) $\in \triangle_{map}$ iff A $\in$ Args1 and r: $l_1 \wedge \ldots \wedge l_k \Rightarrow a$ is a leaf node in A*
- *rule($\mathcal{A}$,r) $\in \triangle_{map}$ iff A $\in$ Args1 and r: $l_1 \wedge \ldots \wedge l_k \Rightarrow l$ is a node in A*
- *rule_head($\mathcal{A}$,r,h) $\in \triangle_{map}$ iff rule($\mathcal{A}$,r) $\in \triangle_{map}$, head(r) = $\{h\}$*
- *rule_body($\mathcal{A}$,r,b) $\in \triangle_{map}$ iff rule($\mathcal{A}$,r) $\in \triangle_{map}$, b $\in$ body(r)*
- *(action($\mathcal{A}$,a) $\wedge$ goal($\mathcal{A}$,g) $\wedge$ rule_body($\mathcal{A}$,r,a) $\wedge$ rule_head($\mathcal{A}$,r,g) $\rightarrow$ action_goal($\mathcal{A}$,a,g)) $\in \triangle_{map}$*
- *(action($\mathcal{A}$,a) $\wedge$ goal($\mathcal{A}$,g) $\wedge$ rule_body($\mathcal{A}$,r,a) $\wedge$ rule_head($\mathcal{A}$,r,h) $\wedge$ (h $\neq$ g) $\wedge$ rule_body($\mathcal{A}$,r$'$,h) $\wedge$ rule_head($\mathcal{A}$,r$'$,g) $\rightarrow$ action_goal($\mathcal{A}$, a, g)) $\in \triangle_{map}$*

Note that the last two rules allow inference of action goal pairs so that one can valuate the temporal or financial cost or efficacy of an action w.r.t. the immediate (sub)goal realised by the action. In the first case, the action is in the antecedent of a goal realisation rule. In the second case, the action is in the body of an action-effect rule whose head (effect) must be (given proposition 3) in the body of a goal realisation rule.

Construction of AF_2 arguments for evaluation of an AF_1 instrumental argument A, proceeds in two steps. Firstly, numerical valuations of sub-arguments of A are inferred from data of the type *temporal_cost(S,a,g,X)*, where S is the source of the valuation of the temporal cost of action a w.r.t goal g. Then second order rules are used to infer a valuation of A from its sub-argument valuations (each of which may be obtained from a different source). In the following, *tc*, *fc*, *eff* and *gp* respectively denote the criteria temporal cost, financial cost, efficacy and goal priority (the importance of a goal to an agent).

Definition 15. $\triangle_{s\_eval}$ *denotes the set of sub-argument evaluation rules :*

- *action_goal($\mathcal{A}$,a,g) $\wedge$ ρ(S,a,g,X) $\rightarrow$ eval(S,ρ,$\mathcal{A}$,a,X), where $\rho \in$ {tc,fc,eff}*
- *initial_goal($\mathcal{A}$,g) $\wedge$ gp(S,g,X) $\rightarrow$ eval(S,gp,$\mathcal{A}$,g,X)*

Definition 16. *Let ρ denote a constant in $\{tc, fc, eff, gp\}$ and Γ a first order theory. Then $\mathcal{D}$ is the following set of Γ specific full-argument evaluation rules.* $d_\rho(\Gamma)$: $eval(S_1, \rho, \mathcal{A}, l_1, X_1)$, ..., $eval(S_n,\rho,\mathcal{A},l_n, X_n) \hookrightarrow eval(\rho, \mathcal{A}, Y)$ *where:*

1. $\{eval(S_1, \rho, \mathcal{A}, l_1, X_1) \ldots eval(S_n, \rho, \mathcal{A}, l_n, X_n)\}$ *is the set of all inferences of the form* $\Gamma \vdash_{FOL} eval(S, \rho, \mathcal{A}, l, X)$
2. $\forall jk, j \neq k \rightarrow l_j \neq l_k$
3. *If* $\rho \in \{tc, fc, eff\}$ *then* $Y = \Sigma_{i=1}^{n} X_i$*, else if* $\rho = gp$ *then* $Y = max_{i=1}^{n} X_i$

Notice that the goal priority of an argument is the maximum of the goal priorities of the argument's initial goals. The financial/temporal cost and efficacy valuation of an argument is the sum of the valuations of the action goal pairs in the argument. The above does not represents an exhaustive list of criteria for evaluating the strength of instrumental arguments. Examples of other criteria include the depth of an argument (preferring arguments of lesser depth favours arguments with fewer intermediate subgoals relating actions to an initial goal), the *certainty level* of an argument (the minimum of the weights associated with rules in an argument), and the number of initial goals in an argument (the criterion used in [1, 2, 8]).

In the following definition we define construction of *AF2* arguments from a first order theory Γ, such that:

- $\Gamma \nvdash_{FOL}$
- $\triangle_{map} \subset \Gamma$, i.e., Γ contains a mapping of instrumental arguments to their sub-arguments in $\mathcal{L}2$
- $\triangle_{s\_eval} \subset \Gamma$, i.e., Γ contains the sub-argument evaluation rules defined in def.15
- $\triangle_{dom} \subset \Gamma$ where $\triangle_{dom}$is a set of domain specific facts of the form $gp(S, g, X)$, $fc(S, a, g, X) \ldots$ used together with rules in $\triangle_{s\_eval}$ to infer valuations of the above sub-arguments
- $\mathcal{ACK} \in \Gamma$ where $\mathcal{ACK}$ is the rule:

  $attack(\mathcal{A}_1, \mathcal{A}_2) \wedge eval(P, \mathcal{A}_1, X) \wedge eval(P, \mathcal{A}_2, Y) \wedge (Y < X) \rightarrow defeat$ $(\mathcal{A}_1, \mathcal{A}_2)$

  for inferring arguments with $defeat$ claims from full-argument valuations
- apart from $\mathcal{ACK}$ there exists no other formula ϕ in Γ such that $defeat(X, Y)$ is a predicate in ϕ. This restriction fulfills the requirement on NAFs in definition 4, viz. a. vie. that $defeat(\mathcal{A}_1, \mathcal{A}_2)$ is a claim of an AF_2 argument built from Γ only if (A_1, A_2) is an attack in $AF_1 =$ (*Args1, Attack1)*

Definition 17. *An argument B based on Γ is a pair (Γ', ϕ), where either:*

1. $\Gamma' = \{\phi_1, \ldots, \phi_n\}$ *where* $d_P(\Gamma) \in \mathcal{D}$ *and* $d_P(\Gamma) = \phi_1, \ldots, \phi_n \hookrightarrow \phi$, *or*
2. $\Gamma' = \Gamma_1 \cup \Gamma_2$, *such that:*
   - $\Gamma_1 = \{\phi_1, \ldots, \phi_n\}$ *where for* $i = 1 \ldots n$, ϕ_i *is the claim of an argument of type 1*
   - $\Gamma_2 \subseteq \Gamma$
   - $\Gamma' \vdash_{FOL} \phi$, *and* Γ' *is consistent and set-inclusion minimal*

Example 5. Continuing with example 3 we list in the left hand column of the table below, the claims of *AF2* sub-argument valuations *J0 - J5'* (writing 'e' as shorthand for 'eval') obtained by def.17-2. We assume that the temporal cost of logging on to the internet is negligible, the agent *ag1*'s initial goal of finishing a paper has higher priority than preparing for a journey to Africa, and getting a vaccination at the hospital takes more time than at the doctor which takes more time than at the private clinic. These are inferred from valuation data in $\triangle_{p\_dom}$[2]. In the middle column we list the claims of *AF2* full argument valuations *K0 - K5* that are supported by *J0 - J5'*. Arguments *K0 - K5* are obtained by def.17-1. In the right hand column we list *AF2* arguments *L0 - L4* for *defeat* claims (we write '*d*' instead of *defeat* and show only the K arguments providing support) obtained by def.17-2. Examples of constructed arguments include:

$J0 = (\{\ initial\_goal(\mathcal{A}1,fp)\ ,\ gp(ag1,fp,0.8),\ initial\_goal(\mathcal{A}1,fp) \wedge gp(ag1,fp,0.8) \rightarrow e(ag1,gp,\mathcal{A}1,fp,0.8)\},\ e(ag1,gp,\mathcal{A}1,fp,0.8)\ )$
$K0 = (\{\ e(ag1,gp,\mathcal{A}1,fp,0.8),\ e(ag1,gp,\mathcal{A}1,pja,0.2)\},\ e(gp,\mathcal{A}1,0.8))$
$L0 = (\{\ attack(\mathcal{A}1,\mathcal{A}2),\ e(gp,\mathcal{A}1,0.8)),\ e(gp,\mathcal{A}2,0.2))\} \cup \{\mathcal{ACK}\},\ d(\mathcal{A}1,\mathcal{A}2))$

$J0 = e(ag1,gp,\mathcal{A}1,fp,0.8)$ $J0' = e(ag1,gp,\mathcal{A}1,pja,0.2)$	$K0 = e(gp,\mathcal{A}1,0.8)$	
$J1 = e(ag1,\ gp,\ \mathcal{A}2,\ pja,\ 0.2)$	$K1 = e(gp,\ \mathcal{A}2,\ 0.2)$	$L0 = (K1 \cup K0,d(\mathcal{A}1,\mathcal{A}2))$
$J2 = e(ag1,\ gp,\ \mathcal{A}3,\ fp,\ 0.8)$ $J2' = e(ag1,\ gp,\ \mathcal{A}3,\ pja,\ 0.2)$	$K2 = e(gp,\ \mathcal{A}3,\ 0.8)$	$L1 = (K1 \cup K2,d(\mathcal{A}3,\mathcal{A}2))$
$J3 = e(ag1,\ tc,\ \mathcal{A}1,\ hop,\ 1)$ $J3' = e(ag1,\ tc,\ \mathcal{A}1,\ w,\ 0.5)$	$K3 = e(tc,\ \mathcal{A}1,\ 1.5)$	$L2 = (K3 \cup K4,d(\mathcal{A}2,\mathcal{A}1))$
$J4 = e(ag1,\ tc,\ \mathcal{A}2,\ pc,\ 2)$	$K4 = e(tc,\ \mathcal{A}2,\ 2)$	$L3 = (K3 \cup K5,d(\mathcal{A}3,\mathcal{A}1))$
$J5 = e(ag1,\ tc,\ \mathcal{A}3,\ dr,\ 1.3)$ $J5' = e(ag1,\ tc,\ \mathcal{A}3,\ w,\ 0.5)$	$K5 = e(tc,\ \mathcal{A}3,\ 1.8)$	$L4 = (K4 \cup K5,d(\mathcal{A}2,\mathcal{A}3))$

We now define the binary relation 'conflict' over *wff* of $\mathcal{L}2$. In the first case, two *wff* conflict if they represent two different valuations of the same sub-argument l of an instrumental argument $\mathcal{A}$ (by the same or different sources) w.r.t. the same criterion P. In the second case, two *wff* conflict if they represent two different valuations of the same instrumental argument $\mathcal{A}$ w.r.t the same criterion P. The third case represents two conflicting defeat claims.

[2] Note that temporal valuations are normalised, e.g., if getting a vacination at the hospital takes 120 minutes and at the private clinic 60 minutes, then *tc(S,pc,vac,2)* and *tc(S,hop,vac,1)*.

Definition 18. *Let* ϕ_1 *and* ϕ_2 *be wff of* $\mathcal{L}2$*. Then, conflict(*ϕ_1, ϕ_2*) iff:*

- ϕ_1 = *eval(S, P,* $\mathcal{A}$*, l, X),* ϕ_2 = *eval(S′, P,* $\mathcal{A}$*, l, Y),* $X \neq Y$
- ϕ_1 = *eval(P,*$\mathcal{A}$*,X),* ϕ_2 = *eval(P,*$\mathcal{A}$*,Y),* $X \neq Y$
- ϕ_1 = *defeat(*$\mathcal{A}$*,*$\mathcal{A}'$*),* ϕ_2 = *defeat(*$\mathcal{A}'$*,*$\mathcal{A}$*)*

We define the conflict based rebut and undercut attacks on the set *Args2* of arguments given by def.17, and then define *AF2.*

Definition 19. *For all* (Γ, ϕ)*,* $(\Gamma', \phi') \in$ *Args2,*

- (Γ, ϕ) *rebuts* (Γ', ϕ') *iff conflict(*ϕ, ϕ'*)*
- (Γ, ϕ) *undercuts* (Γ', ϕ') *iff* $\exists \phi'' \in \Gamma'$ *such that conflict(*ϕ, ϕ''*)*

Definition 20. *AF2 = (Args2, Attack2), where for all B,B′* $\in$ *Args2, (B,B′)* $\in$ *Attack2 iff B rebuts B′ or B undercuts B′.*

Example 6. Continuing with example 5, no two sub-argument or full argument valuations conflict. Hence, *AF2* = (*Args2,Attack2)* where *Args2* includes *J0* - *J5′*, *K0* - *K5*, *L0* - *L4* and *Attack2* = $\{$*L0* $\rightleftharpoons$ *L2, L1* $\rightleftharpoons$ *L4*$\}$. The preferred arguments of *AF2* are *J0* - *J5′*, *K0* - *K5* and *L3.*

Example 7. Recall that in e.g.4 two *AF1* arguments *A1* and *A2*, respectively relate medical actions *a1* and *a2* to treatment goal *g*. Suppose sources clinical trial 1 (*ct1*) reporting that *a1* is more efficacious than *a2* w.r.t. *g*, and clinical trial 2 (*ct2*) reporting that *a2* is more efficacious than *a1* w.r.t. *g*. Therefore *AF2* = (*Args2,Attack2)* where:

- *Args2* includes:
  - *J1*, *J2* and *J3* with claims *e(ct1,eff,*$\mathcal{A}$*1,a1,5)*, *e(ct1,eff,*$\mathcal{A}$*2,a2,4)* and *e(ct2, eff,* $\mathcal{A}$*2,a2,6)* respectively
  - The claims of *J1*, *J2* and *J3* respectively support arguments *K1* with claim *e(eff,* $\mathcal{A}$*1, 5)*, *K2* with claim *e(eff,* $\mathcal{A}$*2, 4)*, and *K3* with claim *e(eff,* $\mathcal{A}$*2, 6)*
  - *K1* and *K2*'s claims support argument *L1* with claim *defeat*($\mathcal{A}$1,$\mathcal{A}$2), and *K1* and *K3*'s claims support *L2* with claim *defeat*($\mathcal{A}$2,$\mathcal{A}$1)

- *Attack2* = $\{$*J2* $\rightleftharpoons$ *J3*, *K2* $\rightleftharpoons$ *K3*, *J3* $\rightharpoonup$ *K2*, *J2* $\rightharpoonup$ *K3*, *L1* $\rightleftharpoons$ *L2*, *K2* $\rightharpoonup$ *L2*, *K3* $\rightharpoonup$ *L1*$\}$

4.2 Defining the Argumentation Framework AF_3

We now define an argumentation system instantiating $AF3$. Priority orderings on sources are used to construct arguments for defeats between $AF2$ sub-argument valuations (e.g., *J2* and *J3* in e.g.7). Priority orderings on criteria are used to construct arguments for defeats between $AF2$ arguments with claims of the form $defeat(\mathcal{A}, \mathcal{A}')$ (e.g., *L0* and *L2* in e.g.6). We will consider a set Π of named partial orderings, where if $\wp$ is the name of an ordering in Π, then this is represented by the usual first order reflexivity and transitivity axioms, and formulae of the form $>(\wp, J,\ K)$ interpreted as source (criterion) J is prioritised above source (criterion) K. We now define the language $\mathcal{L}3$, a mapping from $AF2$ arguments to first order formulae in $\mathcal{L}3$, and rules for construction of $AF3$ arguments:

Definition 21. *Let $AF2 = (Args2 = \{B_1, \ldots, B_n\}, Attack2)$. Then:*

- *$\mathcal{L}3$ is any first order logic language whose signature contains the signature of $\mathcal{L}2$, and the set of constants $\mathsf{f\_name}_2(Args2) = \mathcal{B}_1, \ldots, \mathcal{B}_n$.*
- *$\triangle_{e\_arg} = \{attack(\mathcal{B}_1, \mathcal{B}_2)\ |(B_1, B_2) \in Attack2\} \cup \bigcup_{i=1}^{n} m(B_i)$, where:*
  - *If $claim(B) = eval(S, P, \mathcal{A}, l, X)$ then $m(B) = \{eval(\mathcal{B}, S, P, \mathcal{A}, l, X)\}$*
  - *Else if $B = (\{attack(\mathcal{A}_1, \mathcal{A}_2), eval(P, \mathcal{A}_1, X), eval(P, \mathcal{A}_2, Y) \cup \{\mathcal{ACK}\}\}, defeat(\mathcal{A}_1, \mathcal{A}_2))$ then $m(B) = \{defeat(\mathcal{B}, P, \mathcal{A}_1, \mathcal{A}_2)\}$*
  - *Else $m(B) = \emptyset$*
- *Let $\triangle_{po\_arg}$ be the set of rules:*

  $(attack(\mathcal{B}, \mathcal{B}') \wedge eval(\mathcal{B}, S_1,\ P,\ \mathcal{A},\ l,\ X) \wedge eval(\mathcal{B}', S_2,\ P,\ \mathcal{A},\ l,\ Y) \wedge (X \neq Y) \wedge >(\wp, S_1, S_2)) \rightarrow defeat(\mathcal{B}, \mathcal{B}')$

  $(attack(\mathcal{B}, \mathcal{B}') \wedge defeat(\mathcal{B},\ P,\ \mathcal{A}_1, \mathcal{A}_2) \wedge defeat(\mathcal{B}', P', \mathcal{A}_2, \mathcal{A}_1) \wedge > (\wp, P, P')) \rightarrow defeat(\mathcal{B}, \mathcal{B}')$

An input theory for constructing *AF3* arguments contains the above mapping $\triangle_{e\_arg}$ of *AF2* arguments, a set Π of named orderings on criteria and sources, and the rules $\triangle_{po\_arg}$ for construction of *AF3* arguments. We also assume the restriction (for the same reason as outlined in section 4.1 for an input theory for constructing *AF32* arguments) that the predicate $defeat(X, Y)$ is only in formulae in $\triangle_{e\_arg} \cup \triangle_{po\_arg}$.

Definition 22. *Let Γ be a first order theory such that $\Gamma \nvdash_{FOL} \bot$ and $(\triangle_{e\_arg} \cup \Pi \cup \triangle_{po\_arg}) \subseteq \Gamma$. An argument C based on Γ is a pair (Γ', ϕ), where $\Gamma' \subseteq \Gamma$, $\Gamma' \vdash_{FOL} \phi$ and Γ' is set inclusion minimal.*

Definition 23. *Let AF3 = (Args3, Attack3) where Args3 is the set of all arguments given by def.22, and $\forall C, C' \in$ Args3, $(C, C') \in$ Attack3 iff claim(C) $=$ defeat$(\mathcal{B}, \mathcal{B}')$ and claim(C$'$) $=$ defeat$(\mathcal{B}', \mathcal{B})$.*

Note that no $AF3$ argument attacks another under the conditions that there is only a single criterion ordering and a single source ordering, and no source provides more than one valuation of a sub-argument. Suppose the latter was not satisfied. Then we would have $eval(\mathcal{B}, S_1, \mathrm{P}, \mathcal{A}, l, X)$ and $eval(\mathcal{B}', S_1, \mathrm{P}, \mathcal{A}, l, Y)$, $X \neq Y$ and $> (\wp, S_1, S_1)$ (by reflexivity of $>$) supporting claims $defeat(\mathcal{B}, \mathcal{B}')$ and $defeat(\mathcal{B}', \mathcal{B})$. If the above conditions are not satisfied, then one might need to determine preferences amongst mutually attacking C arguments, which would require construction of *AF4* arguments for preferences amongst criterion/source orderings and sub-argument valuations from a single source.

Definition 24. *Let AF1, AF2 and AF3 = (Args3, Attack3) be defined as in definitions 12, 20 and 23. Let Args3 be defined on the basis of some Γ such that Π contains a single source and a single criterion ordering. Then a nested argumentation framework for agent decision making over instrumental arguments is the triple (AF1,AF2,AF3).*

Example 8. Continuing with example 6, assume a single criterion ordering prioritising goal priority over temporal cost. Then, simply writing this prioritisation in each arguments support, we obtain:
AF3 = (*Args3* = { ({*gp*>*tc*},*defeat(L0,L2)*), ({*gp*>*tc*},*defeat(L1,L4)*) }, ∅).
By def.5:
- *JAF3* = *AF3* and so ***Pf(JAF3)*** = *Args3*

- *JAF2* = *Args2* and *defeat(L0,L2)*, *defeat(L1,L4)*. Hence ***Pf(JAF2)*** now includes *L0*, *L1* and *L3* for claims *defeat*($\mathcal{A}$1,$\mathcal{A}$2), *defeat*($\mathcal{A}$3,$\mathcal{A}$2) and *defeat* ($\mathcal{A}$3,$\mathcal{A}$1).

- *JAF1* is *Args1* and *defeat(A1,A2)*, *defeat(A3,A2)*, *defeat(A3,A1)*. Hence, ***Pf(JAF1)*** = {*A3*}. That is, *A3* is the single preferred instrumental argument given that *A2* is stronger than *A3* is stronger than *A1* on the grounds of temporal cost, but *A3* and *A1* are stronger than *A2* on the grounds of goal priority, where the latter is the preferred criterion.

Example 9. Continuing with example 7, assume a single source ordering *ct1* > *ct2*. Then *AF3* = ({ ({*ct1* > *ct2*},*defeat(J2,J3)*) },∅).
By def.5:
- *JAF3* = *AF3* and so ***Pf(JAF3)*** = ({*ct1* > *ct2*},*defeat(J2,J3)*)

- *JAF2* = (*Args2*,*Defeat2*), where *Defeat2* = *Attack2* - {*(J3,J2)*}. We obtain ***Pf(JAF2)*** = {*J1*, *J2*, *K1*, *K2*, *L1*} where *claim(L1)* = *defeat*($\mathcal{A}$1,$\mathcal{A}$2)

- *JAF1* = {*A1*, *A2*} and *defeat(A1,A2)*. Hence ***Pf(JAF1)*** = *A1*, since although the efficacy of *A2*'s action w.r.t. treatment goal *g* is rated above *A1*'s action by clinical trial 2, the preferred source clinical trial 1 rates *A1*'s action higher than *A2*'s action.

5 Future and Related Work

In this paper we have formalised a framework for nested argumentation, and applied this framework to selection of an agent's preferred instrumental arguments. Future work will more thoroughly investigate properties of nested argumentation frameworks. For example, one might establish the conditions under which arguments are 'objectively' preferred. To illustrate, if *defeat(A2,A1)* and *defeat(A1,A3)* are both based on some criterion *c*, and *defeat(A3,A1)* and *defeat(A1,A2)* are both based on c', then *A2* and *A3* will be preferred irrespective of the ordering of these criteria. One might also consider extending the kinds of 'meta-argumentation' described in frameworks AF_i, $i > 1$. For example, while data concerning the relative strengths of *A1* and *A2* may not be available, a 'transitive' argument for *defeat*($\mathcal{A}1, \mathcal{A}2$) could be constructed from *AF2* arguments for *defeat*($\mathcal{A}1, \mathcal{A}3$) and *defeat*($\mathcal{A}3, \mathcal{A}2$), where the latter two arguments are based on the same criterion. Argumentation over criterion/source orderings will also be investigated. This will require extending *NAF*s to include *AF4* frameworks. For example, a preference for one clinical trial source over another is based on factors including statistical validity, measures taken to eliminate biases e.t.c. This suggests there may be arguments for different orderings on these sources. Finally, application of our work to argumentation-based dialogues [12] would enable agents to engage in the kinds of meta-argumentation described in this paper. For example, an agent justifying to another agent as to why it prefers one argument to another, and this justification itself being challenged. In *deliberation* dialogues, multiple agents cooperate to determine a preferred course of action. A recently proposed model for deliberation [7] describes requirements for communication of arguments for plans of action, and perspectives by which competing arguments are judged. We believe our work has the potential to provide such requirements.

As mentioned in section 1, reasoning about the relative strength of arguments is also explored in [9, 11] in which argument strength is based on rule priorities alone. In value-based argumentation frameworks (VAF) [5] a successful attack (defeat) of one argument by another depends on the comparative strength of the values (analogous to criteria) advanced by the arguments concerned. However, for two arguments that both promote some value v, one cannot defeat the other on the grounds that it promotes v more than the other. Furthermore, VAF is restricted to evaluation of defeats on the basis of value orderings, so that other justifications for defeat are not possible. Also, argumentation over value orderings is not possible.

Section 3 describes how our work on instrumental arguments compares with [1, 2, 8]. To summarise, in our approach arguments more readily instantiate a Dung framework, and preferred arguments are selected on the basis of multiple criteria and sources for valuating the strengths of arguments. Furthermore, as described in example 2, we have defined planning rules so as to 'expose' an instrumental argument's 'potential points of attack'. Future work will further investigate agent argumentation over beliefs and goals and the ways in which these arguments interact with instrumental arguments. Indeed, instrumental arguments can be seen as instantiating a variation on Atkinson et.al's presumptive schema justifying a course of action [4]: *In circumstances R, we should perform action A, whose effects will result in state S which will realise goal G, which promotes some value V.* Arguments attacking an instrumental argument can be seen as instantiating critical questions associated with this schema, e.g.: *does the action have a side effect which demotes some other value?*; *are there alternative ways of realising the same goal?*

Acknowledgements. This work was funded by the European Commission's Information Society Technologies programme, under the IST-FP6-002307 ASPIC project.

References

1. L. Amgoud. A formal framework for handling conflicting desires. In *Proc. 7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'2003)*, pages 552–563, 2003.
2. L. Amgoud and C. Cayrol. On the use of an ATMS for handling conflicting desires. In *Proc. Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pages 175–182, 2004.

3. ASPIC. Deliverable D2.2 - Draft formal semantics for inference and decision-making.
4. K. M. Atkinson, T. J. M. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument for proposals over action. In I. Rahwan, P. Moraitis, and C. Reed, editors, *Proc. First International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2004)*. Springer, 2004.
5. T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
6. P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
7. D. Hitchcock, P. McBurney, and S. Parsons. A framework for deliberation dialogues. In H. V. Hansen et.al, editor, *Proc. Fourth Biennial Conference of the Ontario Society for the Study of Argumentation (OSSA 2001)*, Canada, 2001.
8. J. Hulstijn and L. van der Torre. Combining goal generation and planning in an argumentation framework. In *Proc. 15th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'03)*, 2003.
9. Antonis Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In *Proc. Second international joint conference on Autonomous agents and multiagent systems*, pages 883–890. ACM Press, 2003.
10. J. L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.
11. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.
12. C. Reed and T. J. Norman, editors. *Argument and multi-agent systems - Chapter2. In: Argumentation machines: New frontiers in argument and computation.* Kluwer Academic Publishers, 2004.

Testing Formal Dialectic

Simon Wells and Chris Reed

Division of Applied Computing, University of Dundee, Dundee, UK, DD1 4HN
{swells, chris}@computing.dundee.ac.uk

Abstract. Systems of argumentation or 'computational dialectic' are emerging as a powerful means of structuring inter-agent communication in multi-agent systems. Individual systems of computational dialectic have been suggested and implemented to tackle specific problems but no comprehensive and comparative assessment has been made of such systems. This paper introduces Scenario$_{GC0}$, a framework for the implementation and testing of a wide range of computational dialectic systems. Scenario$_{GC0}$ has a range of benefits for both theoretical and practical work in computational dialectics, including: a means to test arbitrary dialectic systems using a unified knowledge base; a means to determine standard metrics by which dialectic systems can be measured and compared; enabling a body of example dialogue to be assembled for each dialectic system to demonstrate their qualities.

1 Introduction

Certain organisms are used in biological sciences research as models against which to measure new theories. One such organism is *Drosophila Melanogaster* a type of fruit fly [2] which is considered particularly important because much is known about this organism. The body of knowledge about *drosophila* is used as a base line against which to test new theories without extra investment in setting up an experimental structure. New results are thus compared and contrasted against the large body of experimental data already collected. Herbert Simon [20], and later John McCarthy [14] refer metaphorically to the game of Chess as a "drosophila" for AI. In a similar vein McCarthy also proposes the missionaries and cannibals problem as a drosophila for problems in logical AI [15]. The suggestion is that certain classes of problems, puzzles and games can be used to quantify progress in the field of AI overall and to demonstrate individual theories within the field.

This paper presents a drosophila for computational dialectics which we call Scenario$_{GC0}$ together with an implementation framework. The framework enables formal dialectic systems to be rapidly implemented and example dialogues to be produced. This process can be used to investigate the properties of dialectic systems. The results of such an investigation can in turn be used to inform the research, construction and implementation of computational dialectic systems.

Metrics can be identified for dialectic systems and measurements made of the output from running dialectic systems against Scenario$_{GC0}$. This allows the behavior of each system to be examined and quantitative measurements to be

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 74–87, 2006.

derived for those behaviors. The behavior of dialectic systems can thus be measured, compared and evaluated. This fulfills a need in the field of agent communication for a means to evaluate systems of computational dialectic. It also takes the first steps towards building a corpus of example dialogue for each system, generated by an actual implementation of that system. This process enables the identification of the circumstances in which a particular formal dialectic system is most appropriately deployed. It can be used to demonstrate the benefits and efficacy of that formal dialectic system to potential implementors.

2 Problem

Formal dialectic systems were proposed in [7] as practical means to model the interactions between participants in a dialogue in order to examine the situations in which logical fallacies occur. Formal dialectic systems are two-player, turn taking games. The players use their turn to make moves according to the rules of the system. Formal dialectic systems specify the kinds of things that can be said by a participant in a dialogue and when those things can or cannot be said. They don't however make any provision for the propositional content of what is said but concern themselves with the speech acts that are uttered during each players move. Many dialectic systems have been proposed including but not limited to, H [7], DC [10], DL3 [6], PPD [22], R [19] as well as related systems such as the Toulmin Dialogue Game [4], the case study games [11], the eightfold model [12] and variations on existing dialogue games [1]. In addition argumentation, particularly through dialogue games and formal dialectic systems have been proposed as means to structure argumentative dialogue between agents in a MAS [16].

No substantial attempt has been made to establish which system is best for a given application. Many systems have been proposed and many more are possible yet there has been no structured way to approach the specification and implementation of formal dialectic. There has also been no structured approach to establishing the grounds upon which a comparison of systems might be built. Computational testing incorporating the unified specification and implementation of formal dialectic and the production of empirical data from running those systems under known conditions is required. This enables the properties of individual systems of formal dialectic to be determined such that those circumstances to which a given system might be best applied can be established. Because formal dialectic systems make use of but do not provide a formulation for the propositional content of a players moves, the framework that implements a game for testing purposes must supply data that can function as the content of the moves made during a dialogue. Additionally this data must have some provision for argument structures so that the dialogues generated are reminiscent of real-world interactions.

A testing framework should implement a scenario that enables the following: (1) facilitate structured extension and enhancement, (2) enable the automatic generation of arguments with a clear basis for those arguments in the structure of the scenario, (3) produce results that are easily analysed, compared and verified. Further, in a multi-agent system context, a scenario needs to provide a basis for at least the following behaviour: (1) goals that agents can pursue, (2) state changing actions which individual agents can perform. Agents can thus engage in dialectic based inter-agent communication to influence other agents to perform actions in pursuit of goal satisfaction.

3 Scenario

The four colour problem [5]asks whether any map can be coloured using only four colours such that no two neighbouring regions share the same colour. In graph theoretic terms each region of a map may be considered to be a vertice in a graph. It may then be asked whether, given a connected planar graph, only four colours are required to assign each vertice a colour such that no neighbouring vertices share the same colour. For non-planar graphs it may be asked whether the graph can be coloured using n colours such that no neighbouring vertices share the same colour. These type of problems are generally referred to as graph-colouring problems. Scenario$_0$ uses graph colouring problems to provide a basis for an agent society.

The first graph colouring scenario, Scenario$_{GC0}$, is conceived as a testing domain for computational dialectic systems that provides a social context for initiating argumentative dialogue and a knowledge domain to provide a basis for argumentative discourse between agents in a MAS. These properties are leveraged to provide automated, iterative and comparative testing of computational dialectic systems.

The aim is not to provide a solution to a graph colouring problem but to generate test data for computational dialectic systems using a MAS based characterisation of the problem as the basis for argument generation.

3.1 Scenario$_{GC0}$

This scenario is presented as a starting point for examining various types of dialogue including but not limited to information-seeking, persuasion and negotiation. The elements of a core scenario, called Scenario$_{GC0}$, for the graph-colouring problem domain are presented as follows;

Scenario Specific Properties. Each agent in the MAS possesses a colour status. Colours are selected from a fixed pool of colours available to that instance of the scenario. As an initial starting point the pool of colours is fixed at four, red, yellow, green and blue which is reminiscent of the four colour problem. Each agent maintains relationships with a set of other agents in the MAS which are its neighbours. Relationships are defined as an edge joining two vertices. In those cases where two vertices are joined by exactly one edge those vertices are called neighbours. Neighbours are only ever connected directly by one edge although there may be higher order relationships connecting two vertices through other vertices. Relationships are set during system startup and are fixed throughout the duration of the MAS.

Agent Knowledge. At start-up an agent knows only its own colour and a set of neighbouring agents. An agent must therefore research other agents in the MAS in order to increase its knowledge base. To achieve this the agent must engage in dialogue with its neighbours in order to find out the colour properties and relationships of the other agents and identify any conflicts. Conflicts occur when neighbouring agents possess the same colour property. The kind of arguments that an agent can muster and the persuasiveness of those arguments is tied very closely to the knowledge that an agent has. Matters are further complicated because knowledge is uncertain. Whilst an agent can only ever be in a single colour state at any given time that colour state is mutable, it changes with time as the agent determines that another colour state is more suitable. The other agents in the system do not necessarily know that a particular agent has changed colour and might attempt to use information that is no longer accurate. As a result the agents must reason with uncertain and dynamic information in order to achieve their goals.

Conflicts. Conflicts occur when two neighbouring agents share the same colour and is defined as a graph in which two vertices connected by one edge are the same colour. When a conflict occurs it is necessary to resolve the situation. Individual agents have at their disposal the capability to communicate with other agents in order to facillitate a resolution but they have no power to directly influence another agent other than through argumentative dialogue.

Goals. Agents in the MAS maintain goals which pertain directly to the scenario. An agent initially has a single goal, to resolve all conflicts with its neighbours.

Actions. An agent can elect to change its own colour at will should the colour change not bring it into conflict with any of its neighbours. Where there is conflict between agents those agents may elect to change their individual colours in order to remove that conflict. If an agent can change to a free colour, defined as a colour that none of its neighbours currently possesses, then that is the course of action an agent should take. If there are no free colours then an agent might have to change to a colour already possessed by another neighbour even though this will bring it into conflict with that neighbour. In this case the colour change, and resulting movement from a conflict with one neighbour to a conflict with a second neighbours depends upon the argumentation process that has occurred and the agents own internal reasoning.

Conflict Resolution. On discovering a conflict between itself and a neighbouring agent, an agent can make use of the formal-dialectic system at its disposal to bring about a resolution of that conflict. The formal dialectic system might offer various means of conflict resolution involving aspects of, for example, persuasion, negotiation or deliberation. The process of an actual dialogue in terms of the moves that can be made at any given time, the requirements for successful completion of those moves and the effects of a successful move are bound up in the specification of that formal dialectic system as proposed in [23].

Arguments. The agents use arguments to support and justify the actions performed in the system. Formal dialectic systems generally make use of arguments as the content of moves. In this case the arguments are simply propositions that are used to support other propositions and in so doing are used to make a case for the performance or non-performance of some action. Provided that an agent has sufficient knowledge and that the current state of the system is such that an argument can be produced, then whenever a move requires production of some argument the agent should be able to furnish such an argument from its knowledge store.

4 Implementation

A framework to support dialectic testing has been constructed using the Java language and the Jackdaw Agent Framework[9] through the Jackdaw University Development Environment (JUDE). Jackdaw is a lightweight, flexible, industrial-strength agent platform that uses a modular approach to agent development. This enables domain specific functionality to be encapsulated into a module which can be dynamically loaded into a Jackdaw agent at runtime. A Jackdaw module has been implemented that facilitates dialogue between agents in a Jackdaw MAS. The module implements the graph-colouring scenario to enables the automated testing of formal dialectic. The module, named the dialogue manager, is comprised of several data stores and processing components. The data stores include the protocol store, commitment store, dialogue store, template store and knowledge store. The processing components which manipulate the contents of the data stores include an argument manager to facilitate the production of arguments, a protocol manager to govern the process of engaging in argumentative dialogue, and a reasoning component which is embodied in the dialogue manager to facilitate overall control and goal-directed behavior within the module. An overview of the system is shown in 1.

4.1 Protocol Store

The types of communicative acts that an agent can make during a dialogue are regulated by the formal dialectic system in force for that dialogue. The formal dialectic system is stored in a specification format [23] that enables the specification and implementation of arbitrary Hamblin-type formal dialectic systems[7]. Formal dialectic systems have traditionally been specified through lists of locution, commitment, structural and completion rules. Instead, because all a player can do is make moves, moves are made central to the specification of a system and a consideration is made of the effects of making the move and the requirements for doing so. A formal-dialectic system is thus treated merely as a set of moves which the players can make during their turn. Each move is specified in terms of a set of requirements for the move to be legal and the effects of making the move. This has the benefit of enabling systems to be written in a compact format that is both human and machine readable. The range of parameters for

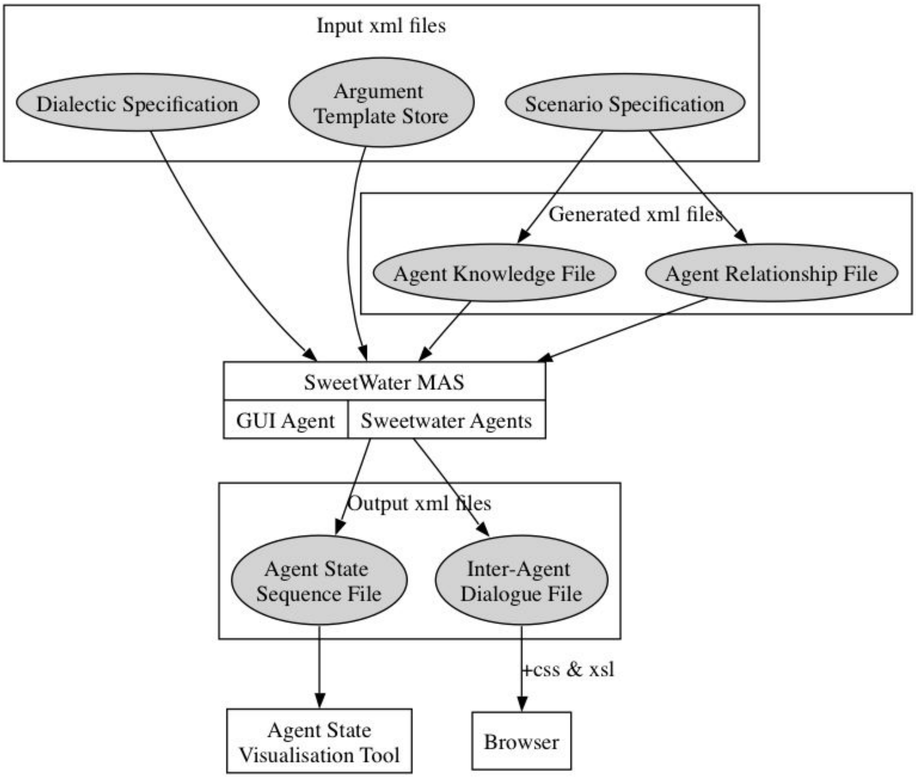


Fig. 1. An overview of the testing framework

move requirements and effects thus far identified enable the following systems to be implemented, H [7], DC [10], DL3 [6], PPD [22], R [19], as well as a myriad of variations on each individual system. Each set of moves that comprise a formal dialectic system is stored in an xml file. Agents can load systems dynamically at runtime to enable the rapid development, implementation and evaluation of new systems.

4.2 Dialogue Store

The dialogue store maintains the transcripts of each dialogue that an agent engages in. This is required for two reasons. Firstly, to fulfill the purposes of a testing framework, It is necessary that not only are results produced by the system but that the process of achieving those results is both clearly represented and easily comprehended. Secondly, the rules of many formal dialectic systems rely upon being able to verify earlier dialogical events and the process of producing arguments can be simplified if a store of arguments that have already been used is maintained. Thus the dialogue store maintains responsibility not only for enabling an agent to maintain some memory of earlier dialogical events but also for the production of transcripts of each dialogue for use in the analysis and verification process.

4.3 Commitment Store

The commitment store maintains a record of the commitments of the participants during a dialogue. In addition to storing the current commitments of participants, in accordance with the current formal dialectic rules, the commitment store enables earlier commitments which might have since been retracted to be examined. This enables rules to be formulated that govern whether a move is admissible based upon whether it has ever been committed to in the past.

4.4 Protocol Manager

The protocol manager utilises the protocol, dialogue and commitment stores in order to govern the process of engaging in argumentative dialogue. This process involves determining the current set of legal moves that can be made dependent upon the moves allowed by the system, earlier moves in the current dialogue and the state of the player's commitment stores.

4.5 Knowledge Store

The knowledge store contains the agents beliefs about agent properties and relationships. This knowledge is represented in an xml file which enables agent knowledge to be organised in a top-down fashion. This facilitates the structured expansion of the concepts that an agent can store as enhanced scenarios are implemented whilst maintaining a strong correspondence between an agents knowledge and the scenario. The knowledge store is essentially a frame-based implementation of knowledge representation. The following tags, once instantiated, are sufficient to record all the information that an agent needs to know to operate successfully in Scenario$_{GC0}$:

⟨*name*⟩is a given agent's unique identifier for some other agent in the MAS.
⟨*colour*⟩is the status of the colour property for the agent indicated by the name tag.
⟨*neighbour_of*⟩records an agent's neighbour. Each neighbour of each agent is recorded using this tag.

The knowledge store provides an interface that enables other components within the agent, such as the argument manager, to retrieve and make use of information. The core of this interface are the getConcept and checkCondition methods. The getConcept method is used to retrieve a proposition from the knowledge store and the check condition method is used to verify that the store contains a particular concept. An agent's knowledge is expanded by adding new tagged data to the store or by specifying methods that process the existing information to extract new concepts. For example the number of conflicts that agent$_1$ has is not stored explicitly but can be calculated by counting the number of neighbours of agent$_1$ who have the same colour state as agent$_1$.

4.6 Template Store

The template store contains *argument templates*. Argument templates can be characterised as semi-instantiated argumentation schemes. Argumentation schemes are traditionally used to capture stereotypical patterns of reasoning and have been used in argument analysis [17, 21] and argument generation [18, 3]. Argument templates are less abstract than argumentation schemes. Templates specify the form of possible arguments within a given knowledge domain whereas schemes are concerned with the form of arguments regardless of the actual content of the arguments. Agents construct arguments by completing a template from the template store with propositional content garnered from the knowledge store to produce an argument instantiation. A template consists of a number of components; a conclusion, a set of one or more premises and a warrant relating the premises to the conclusion. Each component in a template may take one of three forms; static, dynamic or conditional. Static components are expressed in the form of propositions which do not change with respect to the agent's knowledge store. Dynamic components relate to concepts which can be extracted from the knowledge store, the exact values of which vary over time as the agent learns of changes in its situation. Conditional components specify knowledge store concepts which must have particular values. Each component also has a name which corresponds to a concept in the knowledge base. When completing an argument template the getConcept method of the knowledge store is called with the type and name as parameters. This approach enables agents to construct arguments using the dynamic information that they have gathered during interactions with other agents in the system whilst maintaining strict control over the structure of said arguments. The aim here is not to implement a comprehensive model of argument generation but to enable dynamic arguments to be generated in a very controlled manner, to produce known inputs for the process of testing dialectic systems.

Argument templates are specified in an xml document that allows the following tags:

⟨*template*⟩is the name for this template
⟨*scheme*⟩is the scheme associated with this template
⟨*conc type=”type” name=”name”*⟩is the conclusion of the argument
⟨*prem type=”type” name=”name”*⟩is a premise in this argument.
⟨*warrant type=”type” name=”name”*⟩links the premises to the conclusion.

The Templates. Scenario$_{GC0}$ implements several argument templates that build upon the concepts of reducing agent conflicts and increasing the stability of areas of the MAS with respect to agent colour states. For a group of agents of depth d centered around a particular agent, the stability of that group can be measured as the sum of those agents conflicts. The lower the number of conflicts the higher the stability of the agents. The intuition is that agents in more stable areas should make a colour change even if it leads to another conflict if this colour change will increase the stability of the other agent in the less stable area. The templates attempt to capture a path of reasoning from an agent's basic knowledge of colour states and relationships through to a course of action that is based upon that knowledge. As more templates are added to the store agents are able to engage in more varied dialogical behaviours and construct a wider range of arguments. An Araucaria analysis diagram of arguments produced from some of the templates is shown in figure 2. The following template fragment provides an argument for an agent making a colour change based on the fact that they have a conflict and that one agent has more conflicts than the other;

⟨*conc type=”static” name=”colour_change”*⟩You should change colour⟨*/conc*⟩
⟨*prem type=”conditional” name=”conflict_check” /*⟩
⟨*prem type=”conditional” name=”conflict_comparison” /*⟩

The name parameters for each of the premises are passed into the knowledge store and a proposition is returned for each premise which can be expressed as required by the agent as the content of a move. Typically the proposition returned for the conflict_check parameter would ”We are in conflict” and that for the conflict_comparison parameter might be ”You have more conflicts than me”.

Template Chaining. The use of argument templates to provide arguments from the knowledge store suitable for use during a dialogue relies upon the use of *template chaining.* Template chaining involves following the path between templates as required to provide support for a given position. Working from a conclusion, an agent can determine the premises that lend support to that conclusion. The agent can then get supporting data for each premise by treating each premise as the conclusion to a further argument which is in turn supported by other premises. The agent thus finds templates which provide support for each premise and *chains* through the template store instantiating arguments until the premises are based on data which is self-evident from the MAS such as agent colours or relationships. Thus an agent can backwards chain through the templates from the conclusion of an argument to infer the basis for the argument. Likewise an agent can use its observations of agent states and relationships to determine a course of action to follow in pursuit of its goals by forward chaining through the templates.

4.7 The Dialogue Manager

The dialogue manager itself implements the basic reasoning required by the module and coordinates the abilities of the protocol and argument managers. The basic reasoning process that the dialogue manager follows is to find out what options it has at each juncture, e.g. which moves can legally be made. This is achieved by interacting with the protocol manager. The dialogue manager must then determine which moves can be fully instantiated with propositional content from the agents knowledge store, this process is mediated by the argument manager. If an argument is required then the agent utilises the scheme and knowledge stores in an attempt to construct a fully instantiated argument.

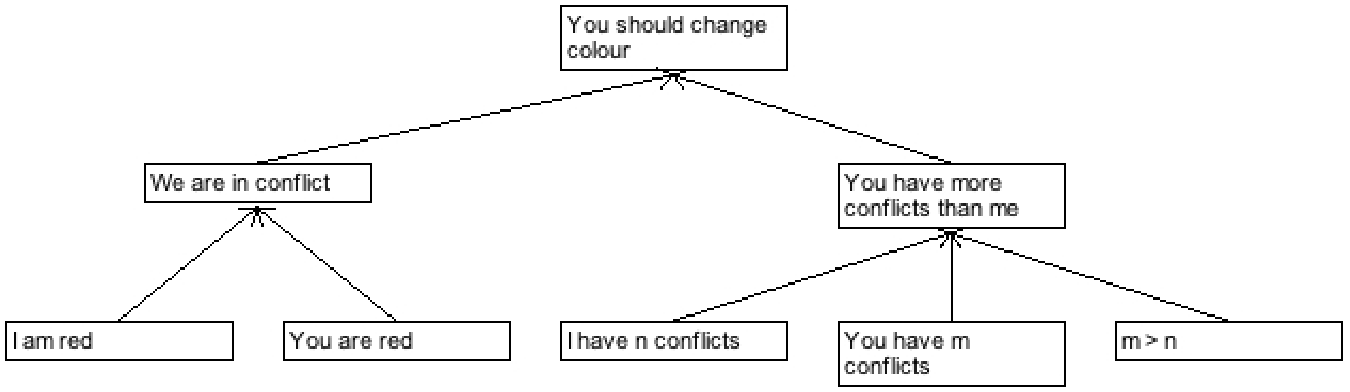


Fig. 2. Instantiations of the arguments specified for the scenario GC_0

5 Metrics for Computational Dialectics

The testing of systems of computational dialectic leads to greater benefits than merely refining the rules of existing games and producing examples. Metrics can be identified by which each distinct dialectic system can be distinguished and categorised. Metrics fall into two broad categories, *inspection metrics*, those metrics which can be measured through external examination of the rules of a system, and *process metrics*, those metrics which are most easily measured through application of the system and subsequent examination of the results. Many inpection metrics are identified in [13] which sets out 13 desirable qualities of argumentative dialogue systems. These include; statement of purpose, diversity of purposes, inclusiveness, transparency, fairness, clarity of theory, separation of syntax and semantics, rule-consistency, encouragement towards resolution, discouragement of disruption, change of position, and system and computational simplicity.

A number of process metrics have been identified that can be applied to the dialogues produced using a computational dialectic system in order to gain insights into that system. The following list is representative but not exhaustive;

5.1 Simplicity of Representation

Communicative Act. How complex are the communicative acts required by the system?

System. How complex is the computational representation of the rules of the system?

5.2 Efficiency of Process

Communication. Size and number of messages

Computational. How much computational power does the system require?

Optimality. How optimal are dialogue results?

Completion. Do all dialogues complete? Are deadlocks avoided? Does the system specify completion conditions?

Repetition. Is repetition of utterances minimised?

5.3 Flexibility

Data Requirements. How complete does data have to be for a dialogue to reach a satisfactory completion.

5.4 Expressiveness

Evolution. Can participants effect a change of position during a dialogue?

Range. Which profiles of dialogue [8] are permitted or prohibited by the system?

Symmetry. Are the same moves available to each participant? Is this always the case?

5.5 Representativeness

Realism. How representative of real world arguments are the resultant dialogues? This is not a prescription for realism but an objective measurement and comparison.

5.6 Stability

Reproducibility. Given the same general inputs, does the system converge to the same set of results?
Predictability. Can the results of a dialogue be predicted?

The analysis of dialogues generated by a given system can be used to make direct measurements of *some* metrics, for example, aspects of the *efficiency of process[communication]* metric can be examined directly. The average number and size of communicative acts per dialogue can be measured and compared between systems. This is sufficient to enable a quantitative comparison to be made between systems on the basis of communicative efficiency. Other metrics rely upon further knowledge against which to compare results, for example, the *expressiveness[range]* metric relies upon knowledge of the range of possible dialogue profiles. Actual measurements of inspection and process metrics can thus be used to categorise individual dialectic systems. This will enable developers to select dialectic systems for use on the basis of their measured attributes and performance.

6 Results

6.1 An Example DC Dialogue

Agent1 and Agent2 are neighbours who are in the same colour state, hence they are in conflict. Upon discovering this Agent1 initiates a dialogue to resolve the conflict. Because of the colour distribution of their neighbours neither agent has any free colours hence any colour change will result in conflict.

Agent$_1$ Statement(”Agent2 should change colour”)
Agent$_2$ Challenge(”Agent2 should change colour”)
Agent$_1$ Defense(”Agent2 is more stable than Agent1”)
Agent$_2$ Challenge(”Agent2 is more stable than Agent1”)
Agent$_1$ Defense(”Agent2 has less conflicts than Agent1”)
Agent$_2$ Challenge(”Agent2 has less conflicts than Agent1”)
Agent$_1$ Defense(”Agent2 has 2 conflicts $\wedge$ Agent1 has 3 conflicts”)
Agent$_2$ Statement(”Agent2 should change colour”)

6.2 An Example H Dialogue

Agent$_1$ Statement(”Agent2 should change colour”)
Agent$_2$ Challenge(”Agent2 should change colour”)
Agent$_1$ Support(”Agent2 is more stable than Agent1”)
Agent$_2$ Challenge(”Agent2 is more stable than Agent1”)

Agent$_1$ Support(”Agent2 has less conflicts than Agent1”)
Agent$_2$ Challenge(”Agent2 has less conflicts than Agent1”)
Agent$_1$ Support(”Agent2 has 1 conflict $\wedge$ Agent1 has 3 conflicts”)
Agent$_2$ Statement(”Agent2 should change colour”)

The examples demonstrate the application of the implementation of Scenario$_{GC0}$ to conflict resolution using the dialectic systems DC and H. The implementation enables agents to engage in information seeking dialogues to discover information about their neighbours and to resolve conflicts when they are discovered through persuasion type dialogues.

7 Conclusions

Scenario$_{GC0}$ and the associated implementation framework are a good way to test systems of formal dialectic through computational implementation. As a baseline it provides a simple, structured body of domain knowledge which forms the content of communicative acts within argumentative dialogue. Further the content of communicative acts is tied closely to the structure of the system and the information inherent in that structure. The basic scenario fulfills the needs of a system for automated production of simple dialogue for testing purposes. The simple nature of generated dialogues can be made more complex through the addition of new argumentation templates to the template store, through the addition of new parameters to the system structure and hence the knowledge of each agent, and through the specification and addition to the protocol store of new systems of formal dialectic. The architecture enables a wider field of experimentation than solely testing models of computational dialectic through the replacement of any or all components so a different model of knowledge might easily be incorporated or different model to govern communications. This is in addition to the flexibility of xml based input data providing a simple, efficient and flexible upgrade path for argument templates, agent knowledge and dialectic systems.

Scenario$_{GC0}$ enables the comparative testing of arbitrary systems of formal dialectic using a standardised knowledge base so that differences in results stem from differences in the dialectic system. It has enabled steps to be taken towards a system of standardised metrics for computational dialectics. Finally it facilitates the automated construction of a corpus of computationally generated dialogue which can be used to compare and contrast the performance of different systems of dialectic. Ongoing work with Scenario$_{GC0}$ involves the testing and comparison of a wide range of formal dialectic systems and dialogue games, the identification of further metrics according to which systems of computational dialectics might be classified, the implementation of additional enhanced scenarios and new argumentation templates.

Acknowledgements

This research is funded by EPSRC under the Information Exchange project. Gratitude is expressed to Calico Jack Ltd. for their JackDaw agent framework and JUDE development environment.

References

1. L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the Fourth International Conference on MultiAgent Systems*, 2000.
2. M. Ashburner. *Drosophila: A Laboratory Handbook.* Cold Spring Harbor Laboratory Press, 1989.
3. K. Atkinson, T. Bench-Capon, and P. McBurney. Justifying practical reasoning. In I. Rahwan, P. Moraitis, and C. Reed, editors, *First International Workshop on Argumentation in Multi-Agent Systems*, 2004.
4. T. J. M. Bench-Capon. Specification and implementation of toulmin dialogue game. In *Proceedings of JURIX 98*, pages 5–20, 2001.
5. A. Cayley. Open problem. *Proceedings of the London Mathematical Society*, 9:148, 1878.
6. R. A. Girle. Commands in dialogue logic. *Practical Reasoning: International Conference on Formal and Applied Practical Reasoning, Springer Lecture Notes in AI*, 1996.
7. C. L. Hamblin. *Fallacies.* Methuen and Co. Ltd., 1970.
8. E. C. W. Krabbe. Profiles of dialogue. In J. Gerbrandy, M. Marx, M. de Rijke, and Y. Venema, editors, *JFAK - Essays Dedicated to Johan van Benthem on the occasion of his 50th birthday.* Amsterdam University Press, 1999.
9. Calico Jack Ltd. http://www.calicojack.co.uk, 2005.
10. J. D. Mackenzie. Question begging in non-cumulative systems. *Journal Of Philosophical Logic*, 8:117–133, 1979.
11. N Maudet and F. Evrard. A generic framework for dialogue game implementation. In *Proceedings of the Second Workshop on Formal Semantics and Pragmatics of Dialog*, 1998.
12. P. McBurney, D. Hitchcock, and S. Parsons. The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, 2002.
13. P. McBurney, S. Parsons, and M. Wooldridge. Desiderata for agent argumentation protocols. *Proceedings of the First AAMAS*, pages 402–409, 2002.
14. J. McCarthy. Ai as sport. *Science*, 276(5318):1518–1519, 1997.
15. J. McCarthy. Elaboration tolerance, 1998.
16. S. Parsons and N. R. Jennings. Negotiation through argumentation. In *Proceedings of ICMAS'96*, pages 267–274, 1996.
17. C. Reed and G. Rowe. Araucaria: "software for puzzles in argument diagramming and xml. Technical report, University Of Dundee, 2001.
18. C. Reed and D. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. In I. Rahwan, P. Moraitis, and C. Reed, editors, *First International Workshop on Argumentation in Multi-Agent Systems*, 2004.
19. N. Rescher. *Dialectics.* State University of New York Press, Albany., 1977.
20. H. Simon and J. Schaeffer. The game of chess, 1992.
21. D. Walton. *Argumentation Schemes for Presumptive Reasoning.* Lawrence Erlbaum Associates, 1996.
22. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue.* SUNY series in Logic and Language. State University of New York Press, 1995.
23. S. Wells and C. Reed. Formal dialectic specification. In I. Rahwan, P. Moraitis, and C. Reed, editors, *First International Workshop on Argumentation in Multi-Agent Systems*, 2004.

Formal Handling of Threats and Rewards in a Negotiation Dialogue

Leila Amgoud and Henri Prade

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)–C.N.R.S.
Université Paul Sabatier, 118 route de Narbonne,
31062 Toulouse Cedex 4, France
{amgoud, prade}@irit.fr

Abstract. Argumentation plays a key role in finding a compromise during a negotiation dialogue. It may lead an agent to change its goals/ preferences and force it to respond in a particular way. Two types of arguments are mainly used for that purpose: *threats* and *rewards*. For example, if an agent receives a threat, this agent may accept the offer even if it is not fully "acceptable" for it (because otherwise really important goals would be threatened).

The contribution of this paper is twofold. On the one hand, a logical setting that handles these two types of arguments is provided. More precisely, logical definitions of threats and rewards are proposed together with their weighting systems. These definitions take into account that negotiation dialogues involve not only agents' beliefs (of various strengths), but also their goals (having maybe different priorities), as well as the beliefs about the goals of other agents.

On the other hand, a "simple" protocol for handling such arguments in a negotiation dialogue is given. This protocol shows when such arguments can be presented, how they are handled, and how they lead agents to change their goals and behaviors.

Keywords: Argumentation, Negotiation.

1 Introduction

Negotiation is the predominant interaction mechanism between autonomous agents looking for a compromise. Indeed, agents make offers that they find acceptable and respond to offers made to them.

Recent works on negotiation [2, 3, 4, 6, 7, 9, 10, 11] have argued that argumentation can play a key role in finding the compromise. Indeed, an offer supported by a 'good argument' has a better chance to be accepted, because the argument brings new information possibly ignored by the receiver. If this information conflicts with previous beliefs of the receiver, this agent may even revise its beliefs if it has no strong counter-argument for challenging the information. Moreover, argumentation may constrain the future behavior of the agent, especially if it takes the form of a *threat* or of a *reward*. Such arguments complement more classical arguments, called here *explanatory arguments*, which especially aim at providing reasons for believing in a statement. Even if the interest of using threats and

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 88–103, 2006.

rewards in a negotiation dialogue [7, 12] has been emphasized, there has been almost no attempt at modeling and incorporating them in a formal dialogue.

This paper aims at providing a logical setting which handles these two types of arguments, together with explanatory arguments. More precisely, logical definitions of threats and rewards are proposed together with their weighting systems. These definitions take into account that negotiation dialogues involve not only agents' beliefs (of various strengths), but also their goals (having maybe different priorities), as well as the beliefs about the goals of other agents. This paper provides also a "simple" protocol for handling such arguments in a negotiation dialogue. This protocol shows when such arguments can be presented, how they are handled, and how they lead agents to change their goals and behaviors.

The paper is organized as follows: Section 2 introduces the logical language for describing the mental states of the agents. Sections 3, 4 and 5 introduce resp. the explanatory arguments, the threats and rewards. For each type of argument, logical definitions are given together with their weighting systems. Note that the given definitions enable us to distinguish between what the agent finds rewarding (resp. threatening) for it and what it finds rewarding (resp. threatening) for the other agent. In section 6, a general argumentation system which handles the three types of arguments is presented. Section 7 introduces a negotiation protocol which is based on the notions of threats and rewards, and which show when such arguments can be presented, how they are handled by their receivers, and how they lead agents to change their behaviors. The approach is illustrated in section 8 on the example of a negotiation between a boss and a worker. In section 9, we conclude by comparing our proposal with existing works and by presenting some perspectives.

2 The Mental States of the Agents

In what follows, $\mathcal{L}$ denotes a propositional language, $\vdash$ classical inference, and $\equiv$ logical equivalence. We suppose that we have two negotiating agents: P (for proponent) and O (for opponent).

Each agent has got a set $\mathcal{G}$ of *goals* to pursue, a knowledge base $\mathcal{K}$, gathering the information it has about the environment, and a base $\mathcal{GO}$, containing what the agent believes the goals of the other agent are. $\mathcal{K}$ may be pervaded with uncertainty (the beliefs are more or less certain), and the goals in $\mathcal{G}$ and $\mathcal{GO}$ may not have equal priority.

Thus, each base is supposed to be equipped with a complete preordering $\geq$. Relation $a \geq b$ holds iff a is at least as certain (resp. as preferred) as b. For encoding it, we use the set of integers $\{0, 1, \ldots, n\}$ as a linearly ordered scale, where n stands for the highest level of certainty or importance and 0 corresponds to the complete lack of certainty or importance. This means that the base $\mathcal{K}$ is partitioned and stratified into

$$\mathcal{K}_1, \ldots, \mathcal{K}_n (\mathcal{K} = \mathcal{K}_1 \cup \ldots \cup \mathcal{K}_n)$$

such that all beliefs in $\mathcal{K}_i$ have the same certainty level and are more certain than beliefs in $\mathcal{K}_j$ where j < i. Moreover, $\mathcal{K}_0$ is not considered since it gathers formulas which are totally uncertain, and which are not at all beliefs of the agent. Similarly,

$$\mathcal{GO} = \mathcal{GO}_1 \cup \ldots \cup \mathcal{GO}_n and \mathcal{G} = \mathcal{G}_1 \cup \ldots \cup \mathcal{G}_n$$

such that goals in $\mathcal{GO}_i$ (resp. in $\mathcal{G}_i$) have the same priority and are more important than goals in $\mathcal{GO}_j$ (resp. in $\mathcal{G}_j$ where j < i).

Note that some $\mathcal{K}_i$'s (resp. $\mathcal{G}_i$, $\mathcal{GO}_i$) may be empty if there is no piece of knowledge (resp. goal) corresponding to the level i of certainty (resp. importance).

For the sake of simplicity, in all our examples, we only specify the strata that are not empty. Both beliefs and goals are represented by propositional formulas of the language $\mathcal{L}$. Thus a goal is viewed as a piece of information describing a set of desirable states (corresponding to the models of the associated proposition) one of which should be reached.

3 Explanatory Arguments

Explanations constitute the most common category of arguments. In classical argumentation-based frameworks that can handle inconsistency in knowledge bases, each conclusion is justified by arguments. They represent the reasons to believe in a fact.

3.1 Logical Definition

Such arguments have a *deductive* form. Indeed, from premises, a fact or a goal is entailed. Formally:

Definition 1 (Explanatory argument). *An* explanatory *argument is a pair <H, h> such that:*

1. $H \subseteq \mathcal{K}$,
2. $H \vdash h$,
3. *H is consistent and minimal (for ⊆) among the sets satisfying 1) and 2).*

$\mathcal{A}_e$ *will denote the set of all the explanatory arguments that can be constructed from* $\mathcal{K}$.

Note that the bases of goals are not considered when constructing such arguments (only based on agent's beliefs) in order to avoid *wishful thinking.*

3.2 Strength of Explanatory Arguments

In [1], it has been argued that arguments may have forces of various strengths. These forces will play two roles:

1. they allow an agent to compare different arguments in order to select the 'best' ones,
2. the forces are useful for determining the acceptable arguments among the conflicting ones.

Different definitions of the force of an argument have been proposed in [1]. Generally, this force of an argument can rely on the beliefs from which it is constructed. Explicit priorities between beliefs, or implicit priorities such as specificity, can be the basis for defining the force of an argument. However, different other aspects can be taken into account when defining the force of explanatory arguments. In particular, the length of the argument (in terms of the number of pieces of knowledge involved) may be considered since the shorter is the explanation, the better it is and the more difficult it is to challenge it (provided that it is based on propositions that are sufficiently certain).

When explicit priorities are given between the beliefs, such as certainty levels, the arguments using more certain beliefs are found stronger than arguments using less certain beliefs. The force of an explanatory argument corresponds to the *certainty level* of the less entrenched belief involved in the argument. In what follows, we consider this view of the force. In the case of stratified bases, the force of an argument corresponds to the smallest number of a stratum met by the support of that argument. Formally:

Definition 2 (Certainty level). *Let $\mathcal{K} = \mathcal{K}_1 \cup \ldots \cup \mathcal{K}_n$ be a stratified base, and $H \subseteq \mathcal{K}$.*
The certainty level *of H, denoted Level(H) = min$\{j \mid 1 \leq j \leq n$ such that $H_j \neq \emptyset\}$, where H_j denotes $H \cap \mathcal{K}_j$.*

Note that $<H, h>$ is all the stronger as Level(H) has a large value.

Definition 3 (Force of an explanation). *Let $A = <H, h> \in \mathcal{A}_e$. The* force *of A is Force(A) = Level(H).*

This definition agrees with the definition of an argument as a minimal set of beliefs supporting a conclusion. Indeed, when any member of this minimal set is seriously challenged, the whole argument collapses. This makes clear that the strength of the least entrenched argument fully mirrors the force of the argument whatever are the strengths of the other components in the minimal set. The forces of arguments make it possible to compare any pair of arguments. Indeed, arguments with a higher force are preferred.

Definition 4 (Comparing explanations). *Let $A, B \in \mathcal{A}_e$. A is* preferred *to B ($A \succ_e B$) iff Force(A) > Force(B).*

4 Threats

Threats have a negative flavor and are applied to intend to force an agent to behave in a certain way. Two forms of threats can be distinguished:

i) You should do ‘a’ otherwise I will do ‘b’,
ii) You should not do ‘a’ otherwise I will do ‘b’.

The first case occurs when an agent P needs an agent O to do ‘a’ and O refuses. Then, P threatens O to do ‘b’ which, according to its beliefs, will have bad consequences for O. Let us consider an example.

Example 1. *Let's consider a mother and her child.*

Mother: You should carry out your school work ('a').
Child: No, I don't want to.
Mother: You should otherwise I will not let you go to the party organized by your friend next week-end ('b').

The second kind of threats occurs when an agent O wants to do some action 'a', which is not acceptable for P. In this case, P threatens that if O insists to do 'a' then it will do 'b' which, according to P's beliefs, will have bad consequences for O. The following example from [7] illustrates this kind of threat.

Example 2

Labor union: We want a wage increase ('a').
Manager: I cannot afford that. If I grant this increase, I will have to lay off some employees ('b'). It will compensate for the higher cost entailed by the increase.

4.1 Logical Definition

In all what follows, we suppose that P presents an argument to O. In a dialogue, each agent plays these two roles in turn. For a threat to be effective, it should be painful for its receiver and conflict with at least one of its goals. A threat is then made up of three parts: the *conclusion* that the agent who makes the threat wants, the *threat* itself and finally the *threatened goal*. Moreover, it has an *abductive* form. Formally:

Definition 5 (Threat). *A* threat *is a triple* $<H, h, \phi>$ *such that:*

1. h *is a propositional formula,*
2. $H \subseteq \mathcal{K}$,
3. $H \cup \{\neg h\} \vdash \neg\phi$ *such that* $\phi \in \mathcal{GO}$,
4. $H \cup \{\neg h\}$ *is consistent and* H *is minimal (for set inclusion) among the sets satisfying the above conditions.*

When $\mathcal{GO}$ *is replaced by* $\mathcal{G}$ *in the above definition, one obtains the definition of an "*own-threat*".* $\mathcal{A}_t$ *will denote the set of all threats and own-threats that may be constructed from the bases* $<\mathcal{K}, \mathcal{G}, \mathcal{GO}>$.

With definition 5, the notion of own-threat covers both the own evaluation of P for the threats it receives, and the threats it may construct or imagine against itself from its own knowledge. Note that h may be a proposition whose truth can be controlled by the agent (e.g the result of an action), as well as a proposition which is out of its control. In a negotiation, an agent P may propose an offer x refused by O. In this case, the offer x is seen as an own-threat by O. P then entices O in order to accept the offer otherwise it will do an action which may be more painful for O. Here h is $Accept(x)$.

Definition 5 captures the two forms of threats. Indeed, in the first case (You should do 'a' otherwise I will do 'b'), $h =$ 'a', and in the second case (You should not do 'a' otherwise I will do 'b'), $h = \neg a$. 'b' refers to an action which may be inferred from H. The formal definition of threats is then slightly more general.

Example 3. *As said in example 1, the mother threatens her child not to let him go to the party organized by his friend if he does't finish his school work. The mother is supposed to have the following bases:*
$\mathcal{K}_{Mo} = \{\neg Work \rightarrow \neg Party\}$,
$\mathcal{G}_{Mo} = \{Work\}$,
$\mathcal{GO}_{Mo} = \{Party\}$.
The threat addressed by the mother to her child is formalized as follows: $<\{\neg Work \rightarrow \neg Party\}, Work, Party>$.

Let's now consider another dialogue between a boss and his employee.

Example 4

Boss: You should finish your work today.
Employee: No, I will finish it another day.
Boss: If you don't finish it you'll come this week-end to make overtime.

In this example, the boss has the three following bases:
$\mathcal{K}_{Bo} = \{\neg\ FinishWork \rightarrow Overtime\}$,
$\mathcal{G}_{Bo} = \{FinishWork\}$ *and*
$\mathcal{GO}_{Bo} = \{\neg Overtime\}$.
The threat enacted by the boss is: $<\ \{\neg\ FinishWork\ \rightarrow\ Overtime\}, FinishWork, \neg Overtime>$.

4.2 Strength of Threats

Threats involve goals and beliefs. Thus, the force of a threat depends on two criteria: the *certainty level* of the beliefs used in that threat, and the *importance* of the threatened goal.

Definition 6 (Force of a threat). *Let* $A = <H, h, \phi> \in \mathcal{A}_t$.
The force *of a threat A is a pair* $Force(A) = <\alpha, \beta>$ *s.t:* $\alpha = Level(H)$; $\beta = j$ *such that* $\phi \in \mathcal{GO}_j$.

However, when a threat is evaluated by its receiver (opponent), the threatened goal is in $\mathcal{G}$. In fact, the threatened goal may or may not be a goal of the opponent.

Definition 7 (Force of an own-threat). *Let* $A = <H, h, \phi> \in \mathcal{A}_t$.
The force *of an own-threat A is a pair* $<\alpha, \beta>$ *s.t.* $\alpha = Level(H)$; $\beta = j$ *if* $\phi \in \mathcal{G}_j$ *otherwise* $\beta = 0$.

Intuitively, a threat is strong if, according to the most certain beliefs, it invalidates an important goal. A threat is weaker if it involves beliefs with a low certainty, or if it only invalidates a goal with low importance. In other terms, the force of a threat represents to what extent the agent sending it (resp. receiving it) is certain that it will violate the most important goals of the other agent (resp. its own important goals). This suggests the use of a *conjunctive* combination of the certainty of H and the priority of the most important threatened goal. Indeed, a fully certain threat against a very low priority goal is not a very serious threat.

Definition 8 (Conjunctive combination). *Let A, B* $\in \mathcal{A}_t$ *with* $Force(A)$ = *<α, β> and* $Force(B)$ = *<α', β'>.*
A is stronger *than B, denoted by A* $\succ_t$ *B, iff min(α, β) > min(α', β').*

Example 5. *Assume the following scale* $\{0, 1, 2, 3, 4, 5\}$*. Let us consider two threats A and B whose forces are respectively (α,β) = (3, 2) and (α', β') = (1, 5). In this case the threat A is* stronger *than B since min(3, 2) = 2, whereas min(1, 5) = 1.*

However, a simple conjunctive combination is open to discussion, since it gives an equal weight to the importance of the goal threatened and to the certainty of the set of beliefs that establishes that the threat takes place. Indeed, one may feel less threatened by a threat that is certain but has 'small' consequences, than by a threat which has a rather small plausibility, but which concerns a very important goal. This suggests to use a weighted minimum aggregation as follows:

Definition 9 (Weighted conjunctive combination). *Let A, B* $\in \mathcal{A}_t$ *with* $Force(A)$ = *<α, β>,* $Force(B)$ = *<α', β'>.*

A is stronger *than B, A* $\succ_t$ *B, iff min(max(λ, α), β) > min(max(λ, α'), β'), where λ is the weight that discounts the certainty level component.*

The larger λ is, the smaller the role of α in the evaluation. The conjunctive combination is recovered when the value of λ is minimal.

Example 6. *Assume the following scale* $\{0, 1, 2, 3, 4, 5\}$*. Let us consider two threats A and B whose forces are respectively (α,β) = (5, 2) and (α', β') = (2, 5). Using a simple conjunctive combination, they both get the same evaluation* 2*. Taking λ = 3, we have* $min(max(3, 5), 2)$ *= 2 and* $min(max(3, 2), 5)$ *= 3. Thus B is stronger than A.*

The above approach assumes the commensurateness of three scales, namely the certainty scale, the importance scale, and the weighting scale. This requirement is questionable in principle. If this hypothesis is not made, one can still define a relation between threats.

Definition 10. *Let* $A, B \in \mathcal{A}_t$ *with* $Force(A) = <\alpha, \beta>$ *and* $Force(B) = <\alpha', \beta'>$*.*

A is stronger *than B iff:*

1. *β > β' or,*
2. *β = β' and α > α'.*

This definition also gives priority to the importance of the threatened goal, but is less discriminating than the previous one.

5 Rewards

During a negotiation an agent P can entice agent O in order that it does 'a' by offering to do an action 'b' as a reward. Of course, agent P believes that 'b' will contribute to the goals of O. Thus, a reward has generally, at least from the point of view of its sender, a positive character. As for threats, two forms of rewards can be distinguished:

i) If you do ‘a’ then I will do ‘b’.
ii) If you do not do ‘a’ then I will do ‘b’.

The following example illustrates this idea.

Example 7. *A seller proposes to offer a set of blank CDs to a customer if this last accepts to buy a computer.*

5.1 Logical Definitions

Formally, rewards have an abductive form and are defined as follows:

Definition 11 (Reward). *A* reward *is a triple* $<H, h, \phi>$ *such that:*

1. *h is a propositional formula,*
2. *$H \subseteq \mathcal{K}$,*
3. *$H \cup \{h\} \vdash \phi$ such that $\phi \in \mathcal{GO}$,*
4. *$H \cup \{h\}$ is consistent and H is minimal (for set inclusion) among the sets satisfying the above conditions.*

When $\mathcal{GO}$ is replaced by $\mathcal{G}$ in the above definition, one gets the definition of an own-reward*.*

$\mathcal{A}_r$ will denote the set of all the rewards that can be constructed from $<\mathcal{K}$, $\mathcal{G}$, $\mathcal{GO}>$.

Note that the above definition captures the two forms of rewards. Indeed, in the first case (If you do ‘a’ then I will do ‘b’), $h =$ ‘a’, and in the second case (If you do not do ‘a’ then I will do ‘b’), $h = \neg a$.

Example 8. *Let’s consider the example of a boss who promises one of his employee to increase his salary.*

Boss: You should finish this work (‘a’).
Employee: No I can’t.
Boss: If you finish the work I promise to increase your salary (‘b’).

The boss has the following bases:
$\mathcal{K}_n = \{FinishWork \rightarrow IncreasedBenefit\}$,
$\mathcal{K}_{n-1} = \{IncreasedBenefit \rightarrow HigherSalary\}$,
$\mathcal{G}_n = \{FinishWork\}$ *and*
$\mathcal{GO}_n = \{HigherSalary\}$.
The boss presents the following reward in favor of its request ‘FinishWork’: $<\{FinishWork \rightarrow HighBenefit,\ HighBenefit \rightarrow HighSalary\}$, $FinishWork,\ HighSalary>$.

Threats are sometimes thought as negative rewards. This is reflected by the parallel between the two definitions which basically differ in the third condition.

Remark 1. Let $\mathcal{K}$, $\mathcal{G}$, $\mathcal{GO}$ be the three bases of agent P. If $h \in \mathcal{G} \cup \mathcal{GO}$, $<\emptyset$, h, $h>$ is both a reward and a threat.

The above property says that if h is a *common goal* of the two agents P and O, then $<\emptyset$, h, $h>$ can be both a reward and a threat, since the common goals jointly succeed or fail. This is either both a reward and a own-reward, or a threat or a own-threat for P.

5.2 Strength of Rewards

As for threats, rewards involve beliefs and goals. Thus, the force of a reward depends also on two criteria: the certainty level of its support and the importance of the rewarded goal.

Definition 12 (Force of a reward). *Let A = $<H$, h, $\phi> \in \mathcal{A}_r$.*
The force *of a reward A is a pair $Force(A) = <\alpha, \beta>$ s.t: α = Level(H); β = j such that $\phi \in \mathcal{GO}_j$.*

However, when a reward is evaluated by its receiver (opponent), the rewarded goal is in $\mathcal{G}$. In fact, if the proponent does not misrepresent the opponent's goals, the rewarded goal is a goal of the opponent.

Definition 13 (Force of an own-reward). *Let A = $<H$, h, $\phi> \in \mathcal{A}_t$. The* force *of an own-reward A is a pair $<\alpha$, $\beta>$ s.t. α = Level(H); $\beta = j$ if $\phi \in \mathcal{G}_j$, otherwise $\beta = 0$.*

Example 9. *In example 8, the force of the reward*
$<\{FinishWork \rightarrow HighBenefit, HighBenefit \rightarrow$
$HighSalary\}$, FinishWork, HighSalary$>$ is $<$n-1, n$>$.

A reward is strong when for sure it will contribute to the achievement of an important goal. It is weak if it is not sure that it will help to the achievement of an important goal, or if it is certain that it will only enable the achievement of a non very important goal. Formally:

Definition 14 (Conjunctive combination). *Let A, B be two rewards in $\mathcal{A}_r$ with $Force(A)$ = $<\alpha$, $\beta>$ and $Force(B)$ = $<\alpha$', β'$>$.*
A is preferred *to B, denoted by $A \succ_r B$, iff min(α, β) > min(α', β').*

However, as for threats, a simple 'min' combination is debatable, since it gives an equal weight to the importance of the rewarded goal and to the certainty of the set of beliefs that establishes that the reward takes place. Indeed, one may feel less rewarded by a reward that is certain but has 'small' consequences, than by a reward which has a rather small plausibility, but which concerns a very important goal. This suggests to use a weighted minimum aggregation as follows:

Definition 15 (Weighted conj. combination). *Let $A, B \in \mathcal{A}_r$ with $Force(A)$=$< \alpha, \beta >$ and $Force(B)$ =$< \alpha', \beta' >$.*

$A \succ_r B$ iff $min(max(\lambda, \alpha), \beta) > min(max(\lambda, \alpha'), \beta')$, where λ is the weight that discounts the certainty level component.

The larger λ is, the smaller the role of α in the evaluation. The 'min' combination is recovered when the value of λ is minimal. In some situations, an agent may prefer a reward which is sure, even if the rewarded goal is not very important for it, than an uncertain reward with very 'valuable' consequences. This suggests to use a weighted minimum aggregation giving priority to the certainty component of the force, as follows:

Definition 16. *Let* $A, B \in \mathcal{A}_r$ *with* $Force(A) = <\alpha, \beta>$ *and* $Force(B) = <\alpha', \beta'>$.

$A \succ_r B$ *iff* $min(\alpha, max(\lambda, \beta)) > min(\alpha', max(\lambda, \beta'))$, *where* λ *is the weight that discounts the importance of the goal.*

Finally, as for threats, if there is no commensurateness of the three scales, we can still be able to compare two rewards as follows, in the spirit of definition 15:

Definition 17. *Let* $A, B \in \mathcal{A}_r$ *with* $Force(A) = <\alpha, \beta>$ *and* $Force(B) = < \alpha', \beta' >$.

$A \succ_r B$ *iff:*

1. $\beta > \beta$*' or,*
2. $\beta = \beta$*' and* $\alpha > \alpha$*'.*

This definition also gives priority to the importance of the rewarded goal. In the case of an agent which prefers rewards that are certain even if the rewarded goals are not very important, one can use the following preference relation.

Definition 18. *Let* $A, B \in \mathcal{A}_r$ *with* $Force(A) = <\alpha, \beta >$ *and* $Force(B) = < \alpha', \beta' >$.

$A \succ_r B$ *iff:*

1. $\alpha > \alpha$*' or,*
2. $\alpha = \alpha$*' and* $\beta > \beta$*'.*

6 Argumentation System

Due to the presence of potential inconsistency in knowledge bases, arguments may be conflicting. The most common conflict which may appear between explanatory arguments is the relation of *undercut* where the conclusion of an explanatory argument contradicts an element of the support of another explanatory argument. Formally:

Definition 19. *Let* $<H, h>$, $<H', h'> \in \mathcal{A}_e$. $<H, h>$ $defeats_e$ $<H', h'>$ *iff*

1. $<H, h>$ *undercuts* $<H', h'>$ *and*
2. *not* ($<H', h'> \succ_e <H, h>$)

Two threats may be conflicting for one of the three following reasons:

- the support of an argument infers the negation of the conclusion of the other argument. It occurs when, for example, an agent P threatens O to do 'b' if O refuses to do 'a', and at his turn, O threatens P to do 'c' if P does 'b'.

- the threats support contradictory conclusions. It occurs, for example, when two agents P and O have contradictory purposes.
- the threatened goals are contradictory. Since a rational agent should have consistent goals, $\mathcal{GO}$ should be as well consistent, and thus this arises when two threats are given by different agents.

As for threats, rewards may also be conflicting for one of the three following reasons:

- the support of an argument infers the negation of the conclusion of the other argument. It occurs when an agent P promises to O to do 'b' if O refuses to do 'a'. C, at his turn, promises to P to do 'c' if P does not pursue 'b'.
- the rewards support contradictory conclusions. This kind of conflict has no sense if the two rewards are constructed by the same agent. Because this means that the agent will contribute to the achievement of a goal of the other agent regardless what the value of h is. However, when the two rewards are given by different agents, this means that one of them wants h and the other $\neg h$ and each of them tries to persuade the other to change its mind by offering a reward.
- the rewarded goals are contradictory.

Formally:

Definition 20. *Let $<H, h, \phi>$, $<H', h', \phi'> \in \mathcal{A}_t$ (resp. $\in \mathcal{A}_r$).*
$<H', h', \phi'>$ $defeats_t$ $<H, h, \phi>$ (resp. $<H', h', \phi'>$ $defeats_r$ $<H, h, \phi>$) iff

1. *$H' \vdash \neg h$, or $h \equiv \neg h'$, or $\phi \equiv \neg \phi'$, and*
2. *not ($<H, h, \phi> \succ_t <H', h', \phi'>$) (resp. not ($<H, h, \phi> \succ_r <H', h', \phi'>$))*

It is obvious that explanatory arguments can conflict with threats and rewards. In fact, one can easily challenge an element used in the support of a threat or a reward. An explanatory argument can also conflict with a threat or a reward when the two arguments have contradictory conclusions. Lastly, an explanatory argument may conclude to the negation of the goal threatened (resp. rewarded) by the threat (resp. the reward). Formally:

Definition 21. *Let $<H, h> \in \mathcal{A}_e$ and $<H', h', \phi> \in \mathcal{A}_t$ (resp. $\in \mathcal{A}_r$).*
$<H, h>$ $defeats_m$ $<H', h', \phi>$ iff

1. *$\exists h'' \in H'$ such that $h \equiv \neg h''$ or*
2. *$h \equiv \neg h'$ or*
3. *$h \equiv \neg \phi$.*

Note that the force of the arguments is not taken into account when defining the relation "$defeat_m$". The reason is that firstly, the two arguments are of different nature. The force of explanatory arguments involves only beliefs while the the force of threats (resp. rewards) involves beliefs and goals. Secondly, beliefs have priority over goals since it is beliefs which determine whether a goal is justified and feasible.

Since we have defined the arguments and the conflicts which may exist between them, we are now ready to introduce the framework in which they are handled.

Definition 22 (Argumentation framework). *An* argumentation framework *is a tuple* $\langle \mathcal{A}_e, \mathcal{A}_t, \mathcal{A}_r, defeat_e, defeat_t, defeat_r, defeat_m \rangle$.

Any argument may have one of the three following status: *accepted*, *rejected*, or in *abeyance*. Accepted arguments can be seen as strong enough for having their conclusion, h, not challenged. In case of threats, for instance, an accepted threat should be taken seriously into account as well its logical consequences. Rejected arguments are the ones defeated by accepted one. Rejected threats will not be taken into account since they are too *weak* or not *credible*. The arguments which are neither accepted nor rejected are said in abeyance.

Let us define what is an accepted argument. Intuitively, accepted rewards (resp. threats) are the ones which are not defeated by another reward (resp. threat) or by an explanatory argument. Formally:

Definition 23 (Accepted threats/rewards). *Let* $\langle \mathcal{A}_e, \mathcal{A}_t, \mathcal{A}_r, defeat_e, defeat_t, defeat_r, defeat_m \rangle$ *be an argumentation framework.*

- *The* set of acceptable *threats is* $\mathcal{S}_t = \{A \in \mathcal{A}_t \mid \nexists B \in \mathcal{A}_t$ *(resp.* $\mathcal{A}_e$*),* B $defeats_t$ *(resp.* $defeats_m$*)* $A\}$. *A threat* $A \in \mathcal{A}_t$ *is* acceptable *iff* $A \in \mathcal{S}_t$.
- *The* set of acceptable *rewards is* $\mathcal{S}_r = \{A \in \mathcal{A}_r \mid \nexists\ B \in \mathcal{A}_r$ *(resp.* $\mathcal{A}_e$*),* B $defeats_r$ *(resp.* $defeats_m$*)* $A\}$. *A reward* $A \in \mathcal{A}_r$ *is* acceptable *iff* $A \in \mathcal{S}_r$.

7 Negotiation Protocol

As said in section 2, we suppose that we have two negotiating agents: P and O. Each of them has got a set $\mathcal{G}$ of *goals* to pursue, a knowledge base $\mathcal{K}$, and a base $\mathcal{GO}$, containing what the agent believes the goals of the other agent are. To capture the dialogues between these agents we follow [2] in using a variant of the dialogue system DC introduced by MacKenzie [8]. In this scheme, agents make dialogical moves by asserting facts into and retracting facts from *commitment stores* (CSs) which are visible to other agents. A commitment store CS is organized in two components: $CS.Off$ in which the *rejected offers* by the agent will be stored, and $CS.Arg$ which will contain the different arguments presented by the agent.

In addition to the different bases, each agent is supposed to be equipped with an argumentation system $\langle \mathcal{A}_e, \mathcal{A}_t, \mathcal{A}_r, defeat_e, defeat_t, defeat_r, defeat_m \rangle$. Note that the agent P constructs the arguments from the three following bases: $<\mathcal{K} \cup CS_C.Arg, \mathcal{G}, \mathcal{GO}>$.

The common agreement that negotiation aims to reach can be about a unique object or a concatenation of objects. Let X be the set of all possible offers. X is made of propositions or their negations.

7.1 Dialogue Moves

At each stage of the dialogue a participant has a set of legal moves it can make — making *offers*, accepting or rejecting offers, challenging an offer, presenting arguments, making threats or rewards. In sum, the set of allowed moves is

$\{Offer$, $Accept$, $Reject$, $Challenge$, $Argue$, $Threat$, $Reward\}$. For each move we describe how the move updates the CSs (the update rules), give the legal next steps possible by the other agent (the dialogue rules), and detail the way that the move integrates with the agent's use of argumentation (the rationality rules). In the following descriptions, we suppose that agent P addresses the move to the agent O.

Offer(x) where x is any formula in X. This allows the exchange of offers.

Rationality

- $\exists <H, x, \phi> \in S_r$ and it is an own-reward, and
- $<H, x, \phi> \succ_r <H', x', \phi'> \forall <H', x', \phi'> \in S_r$ and it is an own-reward with $x' \in X$.

In other terms, x is the most own-rewarding offer for the agent proposing it.

Dialogue: The other agent can respond with *Accept*(x), $Refuse(x)$, or $Challenge(x)$.

Update: There is no change.

$Challenge(x)$ where x is a formula in X.

Rationality: There is no rationality condition.

Dialogue: The other player can only *Argue*(S, x) where $<S, x> \in \mathcal{A}_e$, or $Threat(H, x, \phi)$, or *Reward*(H, x, ϕ).

Update: There is no change.

After an offer, an agent can respond with

Accept(x) where $x \in X$.

Rationality: An agent P accepts an offer in one of the three following cases:

1. $\exists <H, x, \phi> \in S_r$ and it is an own-reward, and $<H, x, \phi> \succ_r <H', x', \phi'>$ $\forall <H', x', \phi'> \in S_r$ and it is a own-reward with $x' \in X$, or
2. $\exists <H, x, \phi> \in \mathcal{S}_r$ and $<H, x, \phi> \in CS.Arg(O)$. This means that the agent has received an acceptable reward from the other agent.
3. $\exists <H, x, \phi> \in \mathcal{S}_t$ and $<H, x, \phi> \in CS.Arg(O)$. This means that the agent has been seriously threatened by the other agent.

Dialogue: The other player can make any move except $Refuse$.

Update: There is no change.

$Refuse(x)$ where x is any formula in X.

Rationality: $\exists <H, x, \phi> \in S_t$ and $<H, x, \phi>$ is a own-threat.

Dialogue: The other player can make any move except $Refuse$.

Update: $CS_i.Off(P) = CS_{i-1}.Off(P) \cup \{x\}$.

Argue(A) where $A \in \mathcal{A}_e$, or $A \in \mathcal{A}_t$ or $A \in \mathcal{A}_r$.

Rationality: There is no rationality condition.

Dialogue: The other player can make any move except $refuse$.

Update: $CS_i.Arg(P) = CS_{i-1}.Arg(P) \cup \{A\}$.

$Threat(H, h, \phi)$ where $<H, h, \phi> \in \mathcal{A}_t$.

Rationality: $h \in CS.Off(O)$. This avoids that agents send gratuitous threats.
Dialogue: The other agent can respond with any move.
Update: $CS_i.Arg(P) = CS_{i-1}.Arg(P) \cup \{(H, h, \phi)\}$.

$Reward(H, h, \phi)$ where $<H, h, \phi> \in \mathcal{A}_r$.

Rationality: $h \in CS.Off(O)$. This avoids that agents send gratuitous rewards.
Dialogue: The other agent can respond with any move.
Update: $CS_i.Arg(P) = CS_{i-1}.Arg(P) \cup \{(H, h, \phi)\}$.

8 Illustrative Example

Let us illustrate the proposed framework in a negotiation dialogue between a boss B, and a worker W about finishing a work in time.

The knowledge base $\mathcal{K}_B$ of B is made of the following pieces of information, whose meaning is easy to guess ('overtime' is short for 'ask for overtime'):

$\mathcal{K}_n$ = {person-sick, overtime → finished-in-time, ¬ finished-in-time → penalty, finished-in-time → ¬ penalty, overtime-paid → extra-cost, strike → ¬ finished-in-time ∧ extra-cost}.
$\mathcal{K}_{a_1}$ = {person-sick → late-work},
$\mathcal{K}_{a_2}$ = {late-work ∧ ¬ overtime → ¬ finished-in-time}.

with $a_1 > a_2$. Goals of B are:

$\mathcal{G}_{b_1}$ = {¬ penalty},
$\mathcal{G}_{b_2}$ = {¬ extra-cost} with $b_1 > b_2$.

Moreover, for B,

$\mathcal{GO}_n$ = {overtime-paid},
$\mathcal{GO}_c$ = {¬ overtime}.

On his side, W has the following bases:

$\mathcal{K}_n$ = {overtime → late-work, overtime-paid → get-money},
$\mathcal{K}_{d_1}$ = {late-work ∧ overtime-paid → overtime},
$\mathcal{K}_{d_2}$ = {person-sick → late-work},
$\mathcal{K}_{d_3}$ = {¬late-work},
$\mathcal{K}_{d_4}$ = {¬ overtime-paid → strike},

with $d_1 > d_2 > d_3 > d_4$. Goals of W are

$\mathcal{G}_n$ = {overtime-paid},
$\mathcal{G}_{e_1}$ = {¬ overtime},

Finally, $\mathcal{GO}_f$ = {¬ strike}.

Possible actions (what is called the set of possible offers in the previous approach) for B are X = {overtime, ¬ overtime, overtime-paid, ¬overtime-paid}. Here it's a sketch of what can take place between B and W.

Step 1: B is led to make the move $Offer(overtime)$. Indeed, the agent can construct the following own-reward: <{overtime → finished-in-time, finished-in-time → ¬ penalty}, overtime, ¬penalty>. The force of this reward is <n, b_1>. Regarding ¬*overtime*, it can be checked that is not rewarding, and even threatening due to Th_1 = <{person-sick, person-sick→late-work, late-work ∧ ¬overtime →¬finished-in-time, ¬ finished-in-time →penalty}, ¬overtime, ¬penalty>, with the force <$\min(a_1, a_2)$, b_1>. It can also be checked that *overtime* is most rewarding than the other actions in X.

Step 2: When W receives the command *overtime*, he makes the move $Challenge(overtime)$ because he can construct the own-threat <∅, overtime, ¬overtime>. Moreover, the worker believes that he should't do overtime according to the explanatory argument <{overtime → late-work, ¬ late-work}, ¬overtime>.

Step 3: B makes the move Argue(Th_1) where he makes explicit to W his own-threat Th_1 used in step 1 for deciding his offer.

Step 4: Now W believes that there is effectively 'late-work' because he can construct the following accepted argument: <{person-sick, person-sick → late-work}, late-work>. Then he will suggest the offer 'overtime-paid' (Offer(overtime-paid)) because it is the most rewarding for him.

Step 5: B makes the move 'Refuse(overtime-paid)' since <{overtime-paid → extra-cost}, ¬overtime-paid, ¬extra-cost> is an own-threat for B.

Step 6: W threatens to go on strike. He presents the move $Threat(Th_2)$ with Th_2 = <{¬ overtime-paid → strike}, overtime-paid, ¬strike>.

Step 7: Th_2 is very serious by B. Indeed, two important goals of the agent will be violated if the worker executes that threat: ¬*penalty* and ¬ extra-cost. In this case, B makes the move 'Accept(overtime-paid)' even if it is not acceptable for him.

9 Related Works – Conclusion

In [7], a list of the different kinds of arguments that may be exchanged during a negotiation has been addressed. Among those arguments, there are threats and rewards. The authors have then tried to define how those arguments are generated. They presented that in terms of speech acts having pre-conditions. Later on in [12], a way for evaluating the force of threats and rewards is given. However no formalization of the different arguments has been given, nor how their forces are evaluated, nor how they can be defeated.

In this paper we have presented a logical framework in which the arguments are defined. Moreover, the different conflicts which may exist between these arguments are described. Different criteria for defining the force of each kind of arguments are also proposed. Clearly, one may think of refining the criteria, especially by taking into account the number of threats or rewards induced by an offer, or the number of weak elements in the evaluation of certainty level. Since arguments may be conflicting we have studied their acceptability. We have also shown through a simple protocol how these arguments can be handled in a negotiation dialogue.

An extension of this work will be to study more deeply the notion of acceptability of such arguments. In this paper we have presented only the individual acceptability where only the direct defeaters are taken into account. However, we would like to investigate the notion of joint acceptability as defined in [5] in classical argumentation.

References

1. L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Int. J. of Automated Reasoning*, 29:125–169, 2002.
2. L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the International Conference on Multi-Agent Systems*, pages 31–38, Boston, MA, 2000.
3. L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In *Proceedings of the 14th European Conference on Artificial Intelligence*, 2000.
4. L. Amgoud and H. Prade. Reaching agreement through argumentation: A possibilistic approach. In *9 th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 194–201, Whistler, Canada, 2004.
5. 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.
6. A. Kakas and P. Moraitis. Argumentative deliberation for autonomous agents. In *Proceedings of the ECAI'02 Workshop on Computational Models of Natural Argument (CMNA'02)*, pages 65–74, 2002.
7. S. Kraus, K. Sycara, and A. Evenchik. *Reaching agreements through argumentation: a logical model and implementation*, volume 104. Artificial Intelligence, 1998.
8. J. MacKenzie. Question-begging in non-cumulative systems. *Journal of philosophical logic*, 8:117–133, 1979.
9. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261—292, 1998.
10. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge engineering review*, 2004.
11. I. Rahwan, L. Sonenberg, and F. Dignum. Towards interest-based negotiation. In *AAMAS'2003*, 2003.
12. S. D. Ramchurn, N. Jennings, and C. Sierra. Persuasive negotiation for autonomous agents: a rhetorical approach. In *IJCAI Workshop on Computational Models of Natural Arguments*, 2003.

Argument-Based Negotiation in a Social Context*

Nishan C. Karunatillake[1], Nicholas R. Jennings[1], Iyad Rahwan[2], and Timothy J. Norman[3]

[1] School of Electronics and Computer Science, University of Southampton, Southampton, UK
{nnc02r, nrj}@ecs.soton.ac.uk

[2] Institute of Informatics, The British University in Dubai, P.O. Box 502216 Dubai, UAE
(Fellow) School of Informatics, University of Edinburgh, Edinburgh, UK
irahwan@acm.org

[3] Department of Computing Science, University of Aberdeen, Aberdeen, UK
tnorman@csd.abdn.ac.uk

Abstract. *Argumentation-based negotiation* (ABN) provides agents with an effective means to resolve conflicts within a multi-agent society. However, to engage in such argumentative encounters the agents require the ability to generate arguments, which, in turn, demands four fundamental capabilities: a schema to reason in a social context, a mechanism to identify a suitable set of arguments, a language and a protocol to exchange these arguments, and a decision making functionality to generate such dialogues. This paper focuses on the first two issues and formulates models to capture them. Specifically, we propose a coherent schema, based on social commitments, to capture social influences emanating from the roles and relationships of a multi-agent society. After explaining how agents can use this schema to reason within a society, we then use it to identify two major ways of exploiting social influence within ABN to resolve conflicts. The first of these allows agents to argue about the validity of each other's social reasoning, whereas the second enables agents to exploit social influences by incorporating them as parameters within their negotiation. For each of these, we use our schema to systematically capture a comprehensive set of social arguments that can be used within a multi-agent society.

Keywords: Argumentation-based Negotiation, Conflict Resolution.

1 Introduction

Multi-agent systems consist of a collection of autonomous agents that inter-operate within a shared social context and that perform actions to achieve their individual and collective objectives. In such situations, the actions of these individual agents are influenced via two broad forms of motivations. First, the *internal influences* reflect the intrinsic motivations that drive the individual agent to achieve its own internal objectives. Second, as agents reside and operate within a social community, the social context itself influences their actions. Here, we categorise these latter forms as *social influences*.

* The first author is a full time PhD student funded by EPSRC under the project Information Exchange (GR/S03706/01). The authors also extend their gratitude to Pietro Panzarasa, Chris Reed, and Xudong Luo for their thoughts, contributions and discussions.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 104–121, 2006.

Now, in many cases, both forms of influence may be present and they may give conflicting motivations to the individual agent. For instance, an agent may be internally motivated to perform a specific action, whereas at the same time it may also be subject to an external social influence not to perform it. Also an agent may face situations where different social influences motivate it in a contradictory fashion (one to perform a specific action and the other not to). Moreover, in many cases agents have to carry out their actions in environments with incomplete information. Thus, for instance, they may not be aware of the existence of all the social influences that could or indeed should affect their actions and they may also lack the knowledge of certain specific internal influences that drive other agents' behaviours.

In such complex and uncertain environments the need for the agents to interact efficiently and effectively becomes paramount. Given this, *Argumentation-Based Negotiation* (ABN) has been advocated as a promising form of interaction that allows agents to resolve their conflicts within such a society [1, 2]. In more detail, ABN allows agents to exchange additional meta-information such as justifications, critiques, and other forms of persuasive locutions within their interactions. These, in turn, allow agents to gain a wider understanding of the internal and social influences affecting their counterparts, thereby making it easier to resolve certain conflicts that arise due to incomplete knowledge. Furthermore, the negotiation element within ABN also provides a means for the agents to achieve mutually acceptable agreements to the conflicts of interests that they may have in relation to their different influences.

Against this background, the main long term objective of our work is to formulate an agent society that can use such argumentative dialogues to resolve their conflicts. Now, one of the central features required by an agent to engage in such arguments within a society is the capability to generate valid arguments during the discourse of the dialogue. We believe this demands four fundamental capabilities: (i) a schema to reason in social settings; (ii) a mechanism to identify a suitable set of arguments; (iii) a language and a protocol to exchange these arguments; and (iv) a decision making functionality to generate such dialogues. This paper focuses on the first two issues and formulates models to capture them. In so doing, we make three main contributions to the state of the art. *First*, we develop a coherent schema of social influence that provides agents with a means to reason about their actions within a society. Here we use the notion of social commitments as the basic building block for our schema and extend this notion to capture social influences emanating from the roles and relationships of a multi-agent society (see Section 2). *Second*, we illustrate how agents can use our social influence schema to *systematically* derive arguments to negotiate and resolve conflicts within a social context. In so doing, we highlight two major ways that agents can use this schema. The first allows them to affect each others' decisions indirectly by arguing about the social influences that determine their decisions. The second allows the agents to impact each others' decisions by exploiting social influences as parameters within their negotiations (see Section 3). *Third*, we perform a detailed analysis on how agents can use both these forms of social arguments to resolve conflicts with respect to existing social influences and to negotiate new social influences within a multi-agent society (see Section 4).

2 Capturing Social Influence

As mentioned in Section 1, many different forms of external influence affect the actions that an agent performs within a society. Moreover, these social influences emanate from different elements of the society. In particular, many researchers now perceive a society as a collection of *roles* inter-connected via a web of *relationships* [3,4]. These roles and relationships represent two important aspects of social influence within a society. Specifically, when an agent operates within such a social context, it may assume certain specific *roles*, which will, in turn, guide the actions it performs. In a similar manner, the *relationships* connecting the agents enacting their respective roles also influence the actions they perform. To date, an array of existing research, both in social science and in multi-agent systems, attempts to capture the influences of these social factors on the behaviour of the individual. Nevertheless, there is little in the way of consensus at an overarching level [5]. Given this, we progressively introduce what we believe are a minimal set of key notions and explain how we adapt them to build a coherent schema of social influence.

The notion of *social commitment* acts as our basic building block for capturing social influence. First introduced by Castelfranchi [6], it remains simple, yet expressive, and is arguably one of the fundamental approaches for modelling social behaviour among agents in multi-agent systems. In essence a social commitment (SC) is a commitment by one agent to another to perform a stipulated action. More specifically, it is defined as a four tuple relation:

$$\mathrm{SC} = (x, y, \theta, w)$$

where x identifies the agent who is socially commitment to carry out the action (termed the *debtor*), y the agent to whom the commitment is made (termed the *creditor*), θ the associated action, and w the witness of this social commitment.[1] Having defined social commitment, Castelfranchi further explains its consequences for both the agents involved. In detail, a social commitment results in the debtor attaining an *obligation* toward the creditor, to perform the stipulated action. The creditor, in turn, attains certain rights. These include the right to demand or require the performance of the action, the right to question the non-performance of the action, and, in certain instances, the right to make good any losses suffered due to its non-performance. We refer to these rights the creditor gains as the *rights to exert influence*.

This notion of social commitment resulting in an obligation and rights to exert influence, allows us a means to capture social influences between two agents. Thus, when a certain agent is socially committed to another to perform a specific action, it subjects itself to the social influences of the other to perform that action. The ensuing obligation, on one hand, allows us to capture how an agent gets subjected to the social influence of another, whereas, the rights to exert influence, on the other hand, model how an agent gains the ability to exert such social influence upon another. Thereby, the notion of social commitment gives an elegant mechanism to capture social influence resulting between two agents.

[1] In the desire to maintain simplicity within our schema, we avoid incorporating the witness in our future discussions (as Castelfranchi did in his subsequent expositions).

Given this basic building block for modelling social influence between specific pairs of agents, we now proceed to explain how this notion is extended to capture social influences resulting due to factors such as roles and relationships within a wider multi-agent society (i.e., those that rely on the structure of the society rather than the specific individuals who happen to be committed to one another). Specifically, since most relationships involve the related parties carrying out certain actions for each other, we can view a relationship as an encapsulation of social commitments between the associated roles. To illustrate this, consider the relationship between the two roles supervisor and student. For instance, assume the relationship socially influences the student to produce and hand over his thesis to the supervisor in a timely manner. This influence we can perceive as a social commitment that exists between the roles supervisor and student (the student is socially committed to the supervisor to perform the stipulated action). As a consequence of this social commitment, the student attains an obligation toward the supervisor to carry out this related action. On the other hand, the supervisor gains the right to exert influence on the student by either demanding that he does so or through questioning his non-performance. In a similar manner, the supervisor may be influenced to review and comment on the thesis. This again is another social commitment associated with the relationship. In this instance, it subjects the supervisor to an obligation to review the thesis while the student gains the right to demand its performance. In this manner, social commitment again provides an effective means to capture the social influences emanating through roles and relationships of the society (independently of the specific agents who take on the roles).

This extension to the basic definition of social commitment is inspired primarily by the work of Cavedon and Sonenberg [3]. Their work investigates how different social influences emanating via roles and relationships affect the agent's prioritising of goals. However, we refrain from going into the level of modalities of agents (such as goals, beliefs and intentions), but rather stay at the level of actions.[2] The motivation for doing so is twofold. First, our primary interest in this work is to use our model to capture arguments that our agents can use to argue about their actions in an agent society. We also aim to implement this argumentation system and test its performance under various arguing strategies (see Section 6). To this end, we believe a model that focuses on the level of actions, as opposed to goals, beliefs and intentions, will reduce the complexity of our future work. Second, an agent adopting a goal, a belief or an intention can be perceived as an act that it performs. Therefore, focusing on the level of actions loses little in terms of expressiveness.

Our extension also adds certain modifications to the original definition of social commitment. In more detail, we allow a social commitment to exist between roles and not only between agents. The rationale for doing so is to relax the highly constraining requirement present within the Cavedon and Sonenberg model that forces all known roles in a relationship to be filled if any one is occupied. To explain this, consider the previous example relationship between the roles student and supervisor. If we define the social commitment between these two roles it captures the general influence within the relationship. Thus, if some particular person assumes the role of student, he would still

[2] For an extended logical formalism that captures how both the beliefs and intentions, in addition to the goals, of an agent are affected via social influences refer to [4].

be obligated to produce the thesis to his supervisor even though, at the moment, the school has not appointed a specific supervisor to him. Therefore, this subtle deviation allows the agents to maintain a social commitment even though the other party of the relationship is not instantiated.

It is also important to clarify our notion of obligation. Here, we do not strictly adhere to the analysis of Castelfranchi that an honest agent will always gain an internal commitment (resulting in an intention to perform that action) for all its social commitments [6]. On the contrary, in accordance with the works of Cavedon and Sonenberg [3] and Dignum *et al.* [5, 7], we believe that all ensuing obligations resulting due to social commitments exert their own degree of influence upon the individual. Thus, certain social commitments may cause a stronger social influence than others. This is, we believe, an important characteristic in realistic multi-agent societies, where autonomous agents are subjected to contradicting external influences (which may also conflict with their internal influences). Therefore, if an agent is subjected to obligations that either contradict or hinder each other's performance, the agent will make a choice about which obligation to honour. To facilitate this choice, we allow the agents to associate each obligation (resulting due to a social commitment) with its own specific degree of influence. We believe this degree of influence is dependent on two main factors. First, is the relationship that the social commitment is a part of. In more detail, two different social commitments related with the same action, but part of different relationships, can cause obligations with different degrees of influence to the agent. Second, it is also dependent on the associated action. Thus even in the same relationship, certain obligations associated with certain actions may cause a stronger influence than others. Given this descriptive definition and the underlying motivations of our model of social influence, we now formulate these notions to develop a notational representation of the schema.[3]

Definition 1. For $n_A, n_R, n_P, n_\Theta \in \mathbb{N}^+$, let:

- $A = \{a_1, \ldots, a_{n_A}\}$ denote a finite set of agents,
- $R = \{r_1, \ldots, r_{n_R}\}$ denote a finite set of roles,
- $P = \{p_1, \ldots, p_{n_P}\}$ denote a finite set of relationships,
- $\Theta = \{\theta_1, \ldots, \theta_{n_\Theta}\}$ denote a finite set of actions,
- Act : $A \times R$ denote the fact that an agent is acting a role,
- RoleOf : $R \times P$ denote the fact that a role is related to a relationship, and
- In : $A \times R \times P$ denote the fact that an agent acting a role is part of a relationship.

If an agent acts a certain role and that role is related to a specific relationship, then that agent acting that role is said to be part of that relationship (as per Cavedon and Sonenberg [3]):

$$\mathrm{Act}(a, r) \land \mathrm{RoleOf}(r, p) \rightarrow \mathrm{In}(a, r, p) \qquad \text{(Rel. Rule)}$$

Definition 2. Let SC denote a finite set of social commitments and $\mathrm{SC}_\theta^{x \rightarrow y} \in SC$ denote a social commitment where x is the entity (agent or role) acting as the debtor, y is the entity acting as the creditor, and θ is the related action.

[3] However, in the following it is not our objective to provide a formal logical definition to the problem of resolving conflicts among obligations. Such a task is non-trivial and some progress to this end is reported in the works of Torre & Tan [8] and Ross [9]. However, a detailed discussion is not within the scope of this paper.

A social commitment $SC_\theta^{x \to y}$ will result in the debtor attaining an obligation toward the creditor to perform a stipulated action and the creditor, in turn, attaining the right to influence the performance of that action (as per Castelfranchi [6]):

$$\mathrm{SC}_\theta^{x \to y} \to [\mathrm{O}_\theta^{x \to y}]_x^f \wedge [\mathrm{R}_\theta^{y \to x}]_y \,, \qquad \text{(S-Com Rule)}$$

where:

- $[\mathrm{O}_\theta^{x \to y}]_x^f$ represents the obligation that x attains that subjects it to an influence of a degree f (see the previous description in Section 2) toward y to perform θ and
- $[\mathrm{R}_\theta^{y \to x}]_y$ represents the right that y attains which gives it the ability to demand, question, and require x regarding the performance of θ.

Definition 3. Let:

- DebtorOf: $(R \ \cup \ A) \times SC$ denote that a role (or an agent) is the debtor in a social commitment,
- CreditorOf: $(R \ \cup \ A) \times SC$ denote that a role (or an agent) is the creditor in a social commitment,
- ActionOf : $\Theta \times SC$ denote that an act is associated with a social commitment, and
- AssocWith : $SC \times P$ denote that a social commitment is associated with a relationship.

If the roles associated with the relationship are both the creditor and the debtor of a particular social commitment, then we declare that social commitment is associated with the relationship (as per Section 2).

Given these definitions, we can capture the social influences within multi-agent systems as a schema. To this end, Figure 1 gives a natural language representation of the schema and a notational representation is captured via formulae (1) through (6). In the following section, we will use this schema to systematically capture the social arguments that agents can use to argue in societies.

Applying Rel. Rule to a society where: $a_i, a_j \in A \wedge r_i, r_j \in R \wedge p \in P$ s.t. Act(a_i, r_i), Act(a_j, r_j), RoleOf(r_i, p), RoleOf(r_j, p) hold true, we obtain:

$$\mathrm{Act}(a_i, r_i) \wedge \mathrm{RoleOf}(r_i, p) \to \mathrm{In}(a_i, r_i, p) \qquad (1)$$

$$\mathrm{Act}(a_j, r_j) \wedge \mathrm{RoleOf}(r_j, p) \to \mathrm{In}(a_j, r_j, p). \qquad (2)$$

Now, consider a social commitment $\mathrm{SC}_\theta^{r_i \to r_j}$ associated with the relationship p in this society. Applying this to Definition 3 we obtain:

$$(\mathrm{DebtorOf}(r_i, \mathrm{SC}) \wedge \mathrm{RoleOf}(r_i, p)) \wedge (\mathrm{CreditorOf}(r_j, \mathrm{SC}) \wedge \mathrm{RoleOf}(r_j, p)) \wedge \mathrm{ActionOf}(\theta, \mathrm{SC}) \to \mathrm{AssocWith}(\mathrm{SC}_\theta^{r_i \to r_j}, p). \qquad (3)$$

Applying the S-Comm rule to $\mathrm{SC}_\theta^{r_i \to r_j}$ we obtain:

$$\mathrm{SC}_\theta^{r_i \to r_j} \to \left[\mathrm{O}_\theta^{r_i \to r_j}\right]_{r_i}^f \wedge \left[\mathrm{R}_\theta^{r_j \to r_i}\right]_{r_j} . \qquad (4)$$

Combining (1), (3) and (4) we obtain:

$$\mathrm{In}(a_i, r_i, p) \wedge \mathrm{AssocWith}(\mathrm{SC}_\theta^{r_i \to r_j}, p) \to \left[\mathrm{O}_\theta^{a_i \to r_j}\right]_{a_i}^f . \qquad (5)$$

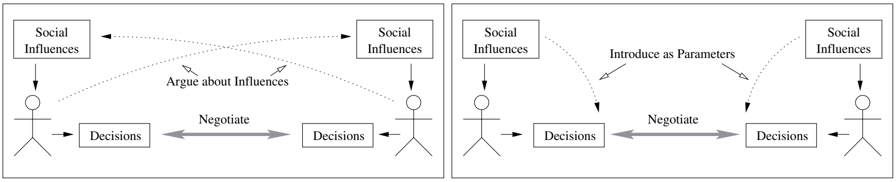
Combining (2), (3) and (4) we obtain:

$$\mathrm{In}(a_j, r_j, p) \wedge \mathrm{AssocWith}(\mathrm{SC}_\theta^{r_i \to r_j}, p) \to \left[\mathrm{R}_\theta^{a_j \to r_i}\right]_{a_j} . \qquad (6)$$

An agent $\boldsymbol{a}_i$ acting the role $\boldsymbol{r}_i$
Leads it to be part of the relationship $\boldsymbol{p}$
With another agent $\boldsymbol{a}_j$ acting the role $\boldsymbol{r}_j$

A social commitment $\boldsymbol{SC}_{\theta}^{r_i \to r_j}$ associated with $\boldsymbol{p}$
- Leads to $\boldsymbol{a}_i$ attaining an obligation $\boldsymbol{O}$ toward $\boldsymbol{r}_j$,
  Which subjects it to an influence of degree $\boldsymbol{f}$
  To perform the action θ
- And, in turn, leads to $\boldsymbol{a}_j$ attaining the right $\boldsymbol{R}$ toward $\boldsymbol{r}_i$
  To demand, question and require the performance of action θ

Fig. 1. Natural Language Representation of the Schema of Social Influence



(a) Socially Influencing Decisions (b) Negotiating Social Influence

Fig. 2. Interplay of Social Influence and Argumentation-Based Negotiation

3 Capturing Social Arguments

When agents operate within a society of incomplete information with diverse and conflicting influences, they may, in certain instances, lack the knowledge, the motivation and/or the capacity to enact all their social commitments. However, to function as a coherent society it is important for these agents to have a means to resolve such conflicts and come to a mutual understanding about their actions. As argued in Section 1, ABN provides one such means. However, to argue in such a society, the agents need to have the capability to first identify the arguments to use. To this end, here we present how agents can use our social influence schema to systematically identify arguments to negotiate within a society. We term these arguments *social arguments*, not only to emphasise their ability to resolve conflicts within a society, but also to highlight the fact that they use the social influence present within the system as a core means in changing decisions and outcomes within the society. More specifically, we have identified two major ways in which social influence can be used to change decisions and outcomes and thereby resolve conflicts between agents. These are depicted in Figure 2 and are described in more detail in the following.

3.1 Socially Influencing Decisions

One way to affect an agent's decisions is by arguing about the validity of that agent's practical reasoning [10, 11]. Similarly, in a social context, an agent can affect another agent's decisions by arguing about the validity of the latter's social reasoning. In more detail, agents' decisions to perform (or not to perform) actions are based on their internal and/or social influences. Thus, these influences formulate the justification (or the reason) behind their decisions. Therefore, agents can affect each other's decisions indirectly by affecting the social influences that determine their decisions (see Figure 2(a)). Specifically, in the case of actions motivated via social influences through the roles and relationships of a structured society, this justification to act (or not to act) flows from the social influence schema (see Section 2). Given this, we can further classify the ways that agents can socially influence each other's decisions into two broad categories:

1. Undercut the opponent's existing justification to perform (or not) an action by disputing certain premises within the schema which motivates its opposing decision.
2. Rebut the opposing decision to act (or not) by,
   (a) Pointing out information about an alternative schema that justifies the decision not to act (or act as the case may be).
   (b) Pointing out information about conflicts that could or should prevent the opponent from executing its opposing decision.

Given this, in the following we highlight how agents can systematically use the social influence schema to identify these possible types of arguments to socially influence each other's decisions. For clarity, we present our arguments both in natural language and using notation. The domain language of our notational representation flows naturally from our schema while the communication language (see Table 1) is inspired from the works of [12], [13], and [14].[4] To denote the arguments we define three additional predicates (to the ones defined in Section 2); namely (i) InfluenceOf(O,f) denotes that f is the degree of influence associated with the obligation O; (ii) $do(a,\phi)$ (or $do(\phi)$ in the abbreviated form) indicating the agent a to perform ϕ (where ϕ can be an action, an obligation, a right to influence, *adopt* a new obligation, or *stop* an existing relationship); (iii) Conflict$(do(\phi_i), do(\phi_j))$ denotes a conflict between $do(\phi_i)$ and $do(\phi_j)$. Finally, in order to illustrate how agents can exploit third party social influences within their arguments, we denote two additional relationships (apart from p defined in Section 2) as p' and p''; the former between the roles r_i and r_k that the agents a_i and a_k hold, and the latter between the roles r_j and r_k that the agents a_j and a_k hold respectively.

1. **Dispute (Dsp.) existing premises to undercut the opponent's existing justification.**
   i. Dsp. a_i is acting role r_i: *ASSERT*$(\neg\text{Act}(a_i, r_i))$.
   ii. Dsp. a_j is acting role r_i: *ASSERT*$(\neg\text{Act}(a_j, r_j))$.
   iii. Dsp. r_i is related to the relationship p: *ASSERT*$(\neg\text{RoleOf}(r_i, p))$.
   iv. Dsp. r_j is related to the relationship p: *ASSERT*$(\neg\text{RoleOf}(r_j, p))$.
   v. Dsp. SC is associated with the relationship p: *ASSERT*$(\neg\text{AssocWith}(\text{SC}_\theta^{r_i \rightarrow r_j}, p))$.
   vi. Dsp. f is the degree of influence associated with O: *ASSERT*$(\neg\text{InfluenceOf}(\text{O}, f))$.
   vii. Dsp. θ is the action associated with O: *ASSERT*$(\neg\text{ActionOf}(\text{O}, \theta))$.
   viii. Dsp. θ is the action associated with R: *ASSERT*$(\neg\text{ActionOf}(\text{R}, \theta))$.

[4] Due to space limitations we intentionally avoid a detailed discussion on the language, the protocol, and the agents' decision making functions of our ABN system. See [15] for further details.

Table 1. High-level description of the communication language

Locution	Description
OPEN-DIALOGUE	Open the dialogue.
CLOSE-DIALOGUE	Close the dialogue.
ASSERT(l)	Assert a certain proposition *l*.
CHALLENGE(l)	Challenge the justification for the proposition *l*.
$\mathit{PROPOSE}(do(l) \Rightarrow do(m))$	Propose the performance of *l* in return for the performance of *m*.
$\mathit{ACCEPT}(do(l) \Rightarrow do(m))$	Accept the performance of *l* in return for the performance of *m*.
$\mathit{REJECT}(do(l) \Rightarrow do(m))$	Reject the performance of *l* in return for the performance of *m*.

2. **Point out (P-o) new premises about an alternative schema to rebut the opposing decision.**
   i. P-o a_i is acting the role r_i: $\mathit{ASSERT}(\mathrm{Act}(a_i, r_i))$.
   ii. P-o a_j is acting the role r_j: $\mathit{ASSERT}(\mathrm{Act}(a_j, r_j))$.
   iii. P-o r_i is related to the relationship p: $\mathit{ASSERT}(\mathrm{RoleOf}(r_i, p))$.
   iv. P-o r_j is related to the relationship p: $\mathit{ASSERT}(\mathrm{RoleOf}(r_j, p))$.
   v. P-o SC is a social commitment associated with the relationship p: $\mathit{ASSERT}(\mathrm{AssocWith}(\mathrm{SC}_\theta^{r_i \to r_j}, p))$
   vi. P-o f is the degree of influence associated with the obligation O: $\mathit{ASSERT}(\mathrm{InfluenceOf}(\mathrm{O}, f))$.
   vii. P-o θ is the action associated with the obligation O: $\mathit{ASSERT}(\mathrm{ActionOf}(\mathrm{O}, \theta))$.
   viii. P-o θ is the action associated with the right R: $\mathit{ASSERT}(\mathrm{ActionOf}(\mathrm{R}, \theta))$.
   ix. P-o a_i's obligation O to perform: $\mathit{ASSERT}(\mathrm{O}_\theta^{a_i \to r_j})$
   x. P-o a_j's right to demand, question and require the action θ: $\mathit{ASSERT}(\mathrm{R}_\theta^{a_j \to r_i})$

3. **Point out conflicts that prevent executing the decision to rebut the opposing decision.**
   (a) Conflicts with respect to O.
      i. P-o a conflict between two different obligations due toward the same role: $\mathit{ASSERT}(\mathrm{Conflict}(do(\mathrm{O}_\theta^{a_i \to r_j}), do(\mathrm{O}_{\theta'}^{a_i \to r_j})))$.
      ii. P-o a conflict between two different obligations due toward different roles: $\mathit{ASSERT}(\mathrm{Conflict}(do(\mathrm{O}_\theta^{a_i \to r_j}), do(\mathrm{O}_{\theta'}^{a_i \to r_k})))$
   (b) Conflicts with respect to R.
      i. P-o a conflict between two different rights to exert influence upon the same role: $\mathit{ASSERT}(\mathrm{Conflict}(do(\mathrm{R}_\theta^{a_j \to r_i}), do(\mathrm{R}_{\theta'}^{a_j \to r_i})))$
      ii. P-o a conflict between two different rights to exert influence upon different roles: $\mathit{ASSERT}(\mathrm{Conflict}(do(\mathrm{R}_\theta^{a_j \to r_i}), do(\mathrm{R}_{\theta'}^{a_j \to r_k})))$
   (c) Conflicts with respect to θ and another action θ' such that (i) θ' is an alternative to the same effect as θ; (ii) θ' either hinders, obstructs, or has negative side effects to θ (refer to [10]).
      $\mathit{ASSERT}(\mathrm{Conflict}(do(\theta), do(\theta')))$

3.2 Negotiating Social Influence

In the previous section, we explored various forms of arguments that agents can use to alter the social influences and thereby change each other's decisions. In this section, we explore a different way in which agents can use social reasoning in negotiation. Here, instead of using social argumentation as a tool to affect decisions, agents use negotiation as a tool for "trading influences". In other words, the social influences are incorporated as additional parameters of the negotiation object itself [16] (see Figure 2(b)). The following presents a list of what we believe to be the most important social arguments that would allow the agents to exploit social influences in such a manner.

4. **Use O as a parameter of negotiation.**
   i. Promise to (or threaten not to) undertake one or many future obligations if the other agent performs (or does not perform) a certain action θ.[5]
   $\mathit{PROPOSE}(\mathit{do}(a_j, \theta) \Rightarrow \mathit{do}(a_i, \text{adopt } \mathrm{O}_{\theta'}^{a_i \to a_j}))$
   $\mathit{PROPOSE}(\mathit{do}(a_j, \theta) \Rightarrow \neg \mathit{do}(a_i, \text{adopt } \mathrm{O}_{\theta'}^{a_i \to a_j}))$
   $\mathit{PROPOSE}(\neg \mathit{do}(a_j, \theta) \Rightarrow \mathit{do}(a_i, \text{adopt } \mathrm{O}_{\theta'}^{a_i \to a_j}))$
   $\mathit{PROPOSE}(\neg \mathit{do}(a_j, \theta) \Rightarrow \neg \mathit{do}(a_i, \text{adopt } \mathrm{O}_{\theta'}^{a_i \to a_j}))$
   ii. Promise to (or threaten not to) honour one or many existing obligations if the other agent performs (or does not perform) a certain action θ: $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \theta) \Rightarrow \pm \mathit{do}(a_i, \mathrm{O}_{\theta}^{a_i \to a_j}))$

5. **Use R as a parameter of negotiation.**
   i. Promise not to (or threaten to) exercise the right to influence one or many existing obligations if the other agent performs (or does not perform) a certain action θ:
   $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \theta) \Rightarrow \pm \mathit{do}(a_i, \mathrm{R}_{\theta'}^{a_i \to a_j}))$

6. **Use third party obligations and rights as a parameter of negotiation.**
   i. Third party obligations
      i. Promise to (or threaten not to) undertake one or more future obligations toward a_k to perform θ', if a_j would (or would not) exercise its right to influence a certain agent a_l to perform θ:
      $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \mathrm{R}_{\theta}^{a_j \to a_l}) \Rightarrow \pm \mathit{do}(a_i, \text{adopt } \mathrm{O}_{\theta'}^{a_i \to a_k}))$
      ii. Promise to (or threaten not to) honour one or more existing obligations toward a_k to perform θ', if a_j would (or would not) exercise its right to influence a certain agent a_l to perform θ:
      $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \mathrm{R}_{\theta}^{a_j \to a_l}) \Rightarrow \pm \mathit{do}(a_i, \mathrm{O}_{\theta'}^{a_i \to a_k}))$
   ii. Third party rights
      i. Promise to (or threaten not to) exercise the right to influence one or many existing obligations toward a_k to perform θ', if a_j would honour its existing obligation to perform θ:
      $\mathit{PROPOSE}(\mathit{do}(a_j, \mathrm{O}_{\theta}^{a_i \to a_j}) \Rightarrow \neg \mathit{do}(a_i, \mathrm{R}_{\theta'}^{a_i \to a_k}))$
      $\mathit{PROPOSE}(\neg \mathit{do}(a_j, \mathrm{O}_{\theta}^{a_i \to a_j}) \Rightarrow \mathit{do}(a_i, \mathrm{R}_{\theta'}^{a_i \to a_k}))$

7. **Use P as a parameter of negotiation.**
   i. Threaten to terminate p (its own relationship with a_j) or p' (a third party relationship that a_i has with a_k), if the agent a_j performs (or does not perform) a certain action θ:
   $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \theta) \Rightarrow \mathit{do}(a_i, \text{stop } p))$
   $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \theta) \Rightarrow \mathit{do}(a_i, \text{stop } p'))$
   ii. Threaten to influence another agent (a_k) to terminate its relationship p'' with a_j, if a_j performs (or does not perform) a certain action θ.
   $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \theta) \Rightarrow \mathit{do}(a_i, \mathrm{R}_{\text{stop } p''}^{a_i \to a_k}))$

In summary, these social arguments allow agents to resolve conflicts in two main ways. The first set of arguments facilitate critical discussion about the social influence schema; thus, these allow the agents to critically question and understand the underlying reasons for each others' action. This form of engagement not only allows the agents to extend their incomplete knowledge of the society, but also provides a means to convince their counterparts to change decisions based on such incomplete information, thereby, resolving conflicts within a society. The second set of arguments allows the agents

[5] To save space, we will denote these four variations as $\mathit{PROPOSE}(\pm \mathit{do}(a_j, \theta) \Rightarrow \pm \mathit{do}(a_i, \theta'))$.

to exploit social influences constructively within their negotiations. Thus, providing agents with additional parameters to influence their counterpart to reach agreements and thereby resolve conflicts through a negotiation encounter. Having systematically captured such social arguments through our schema of social influence, in the following section we present an illustrative case study to analyse their use within ABN to overcome conflicts in multi-agent systems.

4 Arguing with Social Influence

To illustrate how the above set of social arguments can be used to resolve conflicts and change outcomes within a social context, we consider the following case study. The scenario is based on a small community comprising three individual agents (referred to as Andy, Ben, and Carl). Each of these agents has certain specific roles in the scenario and they are inter-connected via a set of relationships. Andy, for instance, has two roles (one as a PhD student and another as a project partner), while Ben has the role of supervisor and Carl has the role project manager. The community has two defined relationships; one between Ben and Andy of type supervisor-student and the second between Carl and Andy of type project manager-project partner. To highlight the use of social arguments we consider the following initial setting where Andy is obliged to perform the following three actions:

θ_1: Obligated toward Ben to *write his thesis*.
θ_2: Obligated toward Ben to *write a journal paper*.
θ_3: Obligated toward Carl to *write a software component*.

Apart from these three obligatory tasks, we also assume that Carl wants Andy to undertake an additional obligation to *integrate the software* (referred to as θ_4) to which Andy has expressed his dissent. In this context, we assume that Andy is only capable of performing one of the above tasks due to time restrictions. Therefore, after prioritising them, Andy chooses to perform action θ_1. In the following we analyse two cases where both Ben and Carl attempt to use the social arguments captured in Section 3 to convince Andy to change his decision. In more detail, the first case analyses Ben's attempt to convince Andy to prioritise writing the journal paper to his thesis. The second case analyses Carl's attempt to negotiate with Andy to undertake the additional obligation of integrating the software. Both of these cases illustrate the use of social arguments to change decisions within a society; the former emphasises their effect in resolving conflicts with respect to existing social influences, while the latter uses them to negotiate new social influences.

4.1 Resolving Conflicts Between Existing Social Influences

The following dialogue sets the scene for the first case by highlighting the constraint and the conflict of interest between the different priorities of both Andy and Ben (i.e., Andy wanting to do θ_1 while Ben wanting Andy to do θ_2):[6]

[6] We choose to denote all argumentative dialogues using natural language, since we believe it allows the reader to easily understand their conceptual differences without worrying about the notational syntax. However, these can be easily encoded in the language highlighted in Table 1 (e.g., see Figure 3).

L1 - Ben: Can you finish the journal paper?
L2 - Andy: No, I can't.
L3 - Ben: Why not?

While L2 revealed the conflict, L3 is an attempt by Ben to identify the reason behind Andy's refusal. This is an important junction of the dialogue for Andy. Here, he can either choose to explain the reasons (thereby answer Ben's question) or challenge Ben's right to question him. First, we will consider the later option where Andy chooses to challenge Ben. This can arise due to a number of reasons. For example, due to incomplete information in the system, Andy may not be aware that θ_2 is an obligatory action. On the other hand, he may be aware that θ_2 is an obligation, but may not be aware that the obligation is toward Ben (thus, he is not convinced of Ben's right to question). Due to any of these reasons Andy could challenge Ben's right to question, which would shift[7] the dialogue toward a critical discussion. The social arguments identified in Section 3 allow Ben to respond appropriately to this form of critical questioning, thus, justify his right to question. The following illustrates an example case where Ben combines social arguments 2.i, 2.ix, and 2.x to retort back to Andy's question:

L4 - Andy: Why do you ask?
L5 - Ben: I am your supervisor and you have an obligation toward me to finish the paper, which gives me the right to question its non-performance.

This form of questioning not only allows Andy to expand his incomplete information of the society, but also provides him with a means to filter out the individuals to which he does not have to justify his non-performance. For instance, if George, a PhD colleague of Andy, played the role of the questioner and asked L3, he will not be able to answer Andy's critical question (L4).

Having analysed how Ben can use the social arguments to respond to Andy's critical questioning, we will now proceed to the next important step in this dialogue. Once questioned and convinced of Ben's right to question, Andy is obliged to give reasons for his non-performance. To illustrate this, consider the following dialogue where Andy uses the social argument 3.a.i within L4 as his reason:

L1 - Ben: Can you finish the journal paper?
L2 - Andy: No, I can't.
L3 - Ben: Why not?
L4 - Andy: I have to finish the thesis, and I can't do two things together.

Having establish the conflict and the reason behind it, it is now up to Ben to convince Andy to change his priorities (i.e., finish the paper before writing the thesis). To achieve this goal, Ben can use a number of different approaches. In the following we will analyse some of these and illustrate how our captured set of social arguments help Ben to achieve his goal:

1. ***Socially Influencing Decisions.***

   One option available to Ben is to focus on the validity of Andy's social reasoning. In other words, Ben can attempt to change Andy's decision (to write the thesis over finishing the paper), by indirectly arguing about the social influences that determined

[7] A change from one dialogue type to another within the same discussion is generally referred to as a *shift* in argumentation theory. However, due to space restrictions, we avoid an expansive discussion about different dialogue types, shifts between them, and fallacies involved when performing such shifts. For a detailed discussion refer to [17, 18].

Let:

- O_1 denote the obligation to perform θ_1 (finishing his thesis) and f_1 its associated degree of influence,
- O_2 denote the obligation to perform θ_2 (write a journal paper) and f_2 its associated degree of influence.

Ben: *OPEN-DIALOGUE*
Andy: *OPEN-DIALOGUE*
Ben: $\mathit{PROPOSE}(do(\theta_2))$
Andy: $\mathit{REJECT}(do(\theta_2))$
Ben: $\mathit{CHALLENGE}(\neg do(\theta_2))$
Andy: $\mathit{ASSERT}(\mathrm{Conflict}(do(\theta_2), do(\theta_1)))$
Ben: $\mathit{ASSERT}(\mathrm{O}_2)$
Andy: $\mathit{ASSERT}(\mathrm{O}_1 \wedge (f_1 > f_2))$
Ben: $\mathit{ASSERT}(\neg(f_1 > f_2) \wedge (f_2 > f_1))$
Andy: $\mathit{ACCEPT}((f_2 > f_1) \wedge do(\theta_2) \wedge \neg do(\theta_1))$
Ben: *CLOSE-DIALOGUE*
Andy: *CLOSE-DIALOGUE*

Fig. 3. Notational representation of the sample ABN dialogue

this. In argumentative dialogue terms, this reflects a shift toward a more persuasive form of a dialogue [17]. Specifically, the social arguments captured in Section 3.1 provide Ben with a number of means to achieve this. For instance, Ben can attempt to identify the motives that prompted Andy to prioritise the thesis over the paper and argue and persuade him to change his motives. The following dialogue illustrates how Ben uses his expert opinion[8] to change Andy's perception of the relative importance of these actions and thereby reach an agreement (see Figure 3 for a notational representation of this using our domain and communication language):

L5 - Ben: But you are obliged to finish the paper.
L6 - Andy: Yes, but I am also obliged to write the thesis and I believe it influences me more than the obligation to finish the journal paper.
L7 - Ben: In my expert opinion, I believe it is more important at this point to finish the paper than the thesis. You should change your opinion.
L8 - Andy: I adhere to your expert opinion, therefore I will finish the paper.

Apart from focusing on obligations and rights, Ben could also socially influence Andy by focusing on the related actions. For instance, Ben could reveal additional information that was not readily available for Andy at the time of prioritising his actions. One way of doing this is to highlight the potential merits of writing the paper before the thesis as follows:

L5 - Ben: If you finish the paper, it will help you to write the thesis since you can reuse the same material.

[8] Argument from expert opinion is a specialised type of argument scheme discussed in depth by Walton [11]. One of the main strengths of our social arguments is to provide a means for the agents to exploit such schemes within a social context.

Ben can do this by using the social argument 3.c.i. He can also emphasise the potential disadvantages of Andy's choice by making use of the social argument 3.c.ii:

L5 - Ben: If you attempt to write the thesis without this paper it will negatively affect the quality of your thesis.

2. ***Negotiating Social Influence.***

   Another option Ben has is to focus on using his existing social influences as parameters to negotiate a mutually acceptable agreement and thereby change Andy's original decision. The social arguments captured in Section 3.2 predominantly facilitate this form of an argumentative dialogue. For instance, Ben can focus on his right to influence the obligatory action as follows:

   L5 - Ben: You are obliged to me to finish the paper and I have a right to demand that you do so.
   L6 - Andy: True, but you also have the right to demand me to write thesis. These rights are in conflict due to time restrictions.
   L7 - Ben: Agreed. I will promise not to exercise my right to demand the thesis provided that you finish the paper.

   Social arguments captured through the schema would facilitate this discussion. Specifically, for L5 Ben uses the arguments 2.ix and 2.x, for L6 Andy uses 2.x and 3.b.i, and for L7 Ben, in turn, uses 5.i to reach a mutually acceptable agreement. Alternatively, Ben can focus on a reciprocal obligation that he has toward Andy and use it as a parameter in negotiation. The following illustrates how Ben uses social arguments 2.x and 4.ii in combination to forward a threat (L5) in his negotiation:

   L5 - Ben: I am obliged to review your thesis. However, if you do so without writing the paper I will neither read nor review your thesis.
   L6 - Andy: In that case I will write the paper first.

4.2 Negotiating New Social Influences

Here we analyse how social arguments provide agents with a means to negotiate new social influences within a society. To this end, we consider a case where Carl is attempting to convince Andy to adopt a new obligation to integrate the software, but Andy is not keen in doing so due to his current time constraints. The following illustrate three possibilities that allow Carl to exploit social influence within his negotiation and in each case we highlight how our social arguments facilitate such an approach:

1. *Exploit existing rights to influence*: Carl can focus his negotiation on his existing right to influence θ_3. More specifically, Carl can point out his existing right to demand the performance of θ_3 and offer to refrain from doing so if Andy adopts the additional obligation. Since adopting an obligation is also an action (see Section 3.2), the social arguments 2.x and 5.i will facilitate this line of negotiation.
2. *Exploit Andy's current situation*: Carl can also make use of Andy's current situation in his negotiation. In more detail, if Carl knows that Andy is pressed for time to write the paper, he can offer to help him achieve this if, in exchange, Andy agrees to do the integration. In this line of argument, Andy is proposing to undertake an

additional obligation (helping Andy write the paper) in exchange for Andy adopting the additional obligation. Carl can achieve this by using the social argument 4.i in his negotiation dialogue.

3. *Trade existing obligations*: Carl can offer to trade with Andy his current obligation to do θ_3 with the new obligation to perform θ_4. This could be a more agreeable solution for both Carl and Andy especially if performing θ_4 is not an immediate requirement. This line of argument is a combination of the above two since here Carl rescinds his rights on θ_3 (using argument 4.i), whilst Andy adopts a new obligation on θ_4 (using argument 5.i).

5 Related Work

As detailed in Section 1, one of the central features required by an agent to argue and resolve its conflicts is its capability to generate valid arguments during the discourse. This area is extensively researched in current ABN literature and a wide variety of approaches have been proposed to model this capability within a computational entity [2]. In the following, we review some of these and place our model in context by highlighting similarities and distinctions with these efforts.

Our work greatly benefits from the approaches used in *argumentation schemes* [11] to systematically identify arguments. In more detail, argumentation schemes capture stereotypical patterns of reasoning upon which communication structures can be built. Increasingly, they are used in computational contexts, including multi-agent systems, since they hold potential for significant improvements in reasoning and communication abilities in such systems [19]. For instance, the recent work of Reed & Walton [19] presents a general framework for specifying such schemes in computational contexts and the work of Atkinson *et al.* [10] uses this in their schematic approach to capture a particular style of dialogue over actions. In a similar manner, we have captured how agents function and reason *within a society* as a schema of social influence and use it to systematically identify social arguments. In systematically categorising our social arguments, we also draw from the logical approaches to ABN [12, 20]. Broadly, these systems formulate an argument as a certain sequence of inferences leading to a logical conclusion and, in turn, allow the agents to construct attacks by either disqualifying one or more of these inferences (undercut) or formulating an alternative series of inference leading to the opposite conclusion (rebut). We follow the same systematic manner in organising the different social arguments identified from our schema.

Another fundamental work in computational argument generation is that of Kraus, Sycara & Evenchik [21]. In essence, their work allows agents to use promises, threats and various forms of appeals during a negotiation encounter. Now, our social arguments, particularly those that allow agents to negotiate social influences, hold certain similarities to their forms of threats, promises and appeals. However, there are two important distinctions. First, their main focus is in formulating interactions between two agents, whereas we expressly take into account the impact of society by way of social commitments. Second, they do not take into account incomplete information between the two agents. Thus, they do not provide agents with a means to resolve conflicts due to such imperfections that are often present with a multi-agent system. In contrast, our social arguments captured in Section 3.1 allow agents to argue about their social influences and overcome such conflicts within a society.

The work of Sierra *et al.* [22] is an important initial attempt to extend the work of Kraus *et al.* to a social context. Similar to our approach (and unlike [21]) they allow agents to argue in social contexts with imperfect information. However, they only consider authority based relationships, which we believe only capture a specialised form of social context (i.e., institutions or formal organisations). Our work, on the other hand, presents a more generic way of capturing social influences of roles and relationships (i.e., using social commitment with different degrees of influence). This not only provides a simple unified mechanism to simulate social contexts with a wide array of relationships exerting different social influences upon the agents, but also allows us to experiment with our agents' ability to argue, negotiate and resolve conflicts in such disparate social systems.

6 Conclusions and Future Work

The long term objective of our work is to formulate an agent society that can use argumentative dialogues to resolve their conflicts. As mentioned in Section 1, we believe this requires agents to have four fundamental capabilities. To this end, this paper addresses the first two issues; namely a schema to reason in a social context and a mechanism to identify a suitable set of arguments. We achieve these aims by (i) developing a coherent schema for agents to function among different social influences and (ii) designing a model that allows agents to systematically use this schema to capture social arguments to negotiate and resolve conflicts within a social context. We also highlighted the two main ways in which social influence and ABN mutually enhance one another in terms of effectively resolving conflicts and demonstrated their operation in an illustrative case study.

In addition to the above two issues, we believe that agents also require a language and a protocol to exchange these arguments, and a decision making functionality to generate such dialogues [15]. As mentioned in Section 3.1, our domain language flows naturally from our schema and the communication language is adapted from the works of Amgoud *et al.* [12] and McBurney *et al.* [14]. In abstract, our protocol has six stages; namely opening, conflict recognition, conflict diagnosis, conflict management, agreement, and closing. Apart from the opening and the closing stages, which provide synchronisation points for the agents, the remaining four comply well with the pragma-dialectics model for critical discussion proposed by van Eemeren and Grootendorst [23]. Furthermore, for each locution type we have defined their respective pre-condition and commitment rules. Finally, we have defined the decision making functions for each of these dialogue moves first, at an abstract level and then in a more domain dependent level. However, since the main objective of this paper is to set the conceptual grounding (and also due to space restrictions) we choose to exclude detailed explanations of these issues from this paper. In future, we aim to expand upon our current implementation by designing different argument selection strategies, thus allow the agents to adopt different tactics in resolving conflicts in a multi-agent community.

References

1. Karunatillake, N.C., Jennings, N.R.: Is it worth arguing? In: Argumentation in Multi-Agent Systems (Proc. of ArgMAS 2004). LNAI 3366, NY, USA, Springer-Verlag (2004) 234–250
2. Rahwan, I., Ramchurn, S.D., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review **18** (2003) 343–375
3. Cavedon, L., Sonenberg, L.: On social commitment, roles and preferred goals. In: Proc. of the third Int. Conf. on Multi-Agent Systems (ICMAS'98). (1998) 80–86
4. Panzarasa, P., Jennings, N.R., Norman, T.J.: Social mental shaping: Modelling the impact of sociality on the mental states of autonomous agents. Computational Intelligence **17** (2001) 738–782
5. Dignum, F., Morley, D., Sonenberg, E.A., Cavedon, L.: Towards socially sophisticated BDI agents. In: Proc. of the fourth Int. Conf. on Multi-agent Systems, Boston, USA (2000) 111–118
6. Castelfranchi, C.: Commitments: From individual intentions to groups and organizations. In: Proc. of the first Int. Conf. on Multi-agent Systems (ICMAS'95), San Francisco, CA (1995) 41–48
7. Dignum, V., Kinny, D., Sonenberg, L.: Motivational attitudes of agents: On desires, obligations and norms. In: Proc. of the second Int. Workshop of Central Eastern Europe on Multi-Agent Systems (CEEMAS'01). Volume 2296., Poland (2001) 61–70
8. Torre van der, L., Tan, Y.H.: Contrary-to-duty reasoning with preference-based dyadic obligations. Annals of Mathematics and Artificial Intelligence **27** (1999) 49–78
9. Ross, A.: Imperatives and logic. Theoria **7** (1941) 53–71
10. Atkinson, K., Bench-Capon, T., McBurney, P.: A dialogue game protocol for multi-agent argument over proposals for action. In: Argumentation in Multi-Agent Systems (Proc. of ArgMAS 2004). LNAI 3366, NY, USA, Springer-Verlag (2004) 149–161
11. Walton, D.N.: Argumentation Schemes for Presumptive Reasoning. Erlbaum, Mahwah, NJ (1996)
12. Amgoud, L., Parson, S., Maudet, N.: Argument, dialogue and negotiation. In Horn, W., ed.: Proc. of the 14^{th} European Conference on Artificial Intelligence (ECAI'00), Berlin (2000) 338–342
13. MacKenzie, J.: Question-begging in non-cumulative systems. Journal of philosophical logic **8** (1979) 117–133
14. McBurney, P., van Eijk, R., Parsons, S., Amgoud, L.: A dialogue-game protocol for agent purchase negotiations. Autonomous Agents and Multi-Agent Systems **7** (2003) 235–273
15. Karunatillake, N.C., Jennings, N.R., Rahwan, I., Norman, T.J.: Arguing and negotiating in the presence of social influences. In: Proc. of the fourth Int. Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'05). LNAI 3690, Budapest, Hungary, Springer-Verlag (2005) 223–235
16. Faratin, P., Sierra, C., Jennings, N.R.: Using similarity criteria to make trade-offs in automated negotiations. Artificial Intelligence **142** (2002) 205–237
17. Reed, C.A.: Dialogue frames in agent communication. In: Proc. of the third Int. Conf. on Multi-Agent Systems. (1998) 246–253
18. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State Univ. of NY (1995)
19. Reed, C.A., Walton, D.N.: Towards a formal and implemented model of argumentation schemes in agent communication. In: Argumentation in Multi-Agent Systems (Proc. of ArgMAS 2004). LNAI 3366, NY, USA, Springer-Verlag (2004) 19–30
20. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation **8** (1998) 261–292

21. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation. Artificial Intelligence **104** (1998) 1–69
22. Sierra, C., Jennings, N.R., Noriega, P., Parsons, S.: A framework for argumentation-based negotiation. In: Proc. of fourth Int. Workshop on Agent Theories Architectures and Languages (ATAL'97), Rhode Island , USA (1998) 167–182
23. Eemeren van, F.H., Grootendorst, R.: Argumentation, Communication, and Fallacies. Lawrence Erlbaum Associates, Inc, Hillsdale NJ (1992)

Practical Strategic Reasoning and Adaptation in Rational Argument-Based Negotiation

Michael Rovatsos[1], Iyad Rahwan[2], Felix Fischer[3], and Gerhard Weiss[3]

[1] School of Informatics, The University of Edinburgh, Edinburgh EH8 9LE, UK
mrovatso@inf.ed.ac.uk
[2] Institute of Informatics, The British University in Dubai, P.O. Box 502216, Dubai, UAE
iyad.rahwan@buid.ac.ae
[3] Department of Informatics, Technical University of Munich, 85748 Garching, Germany
{fischerf, weissg}@cs.tum.edu

Abstract. Recent years have seen an increasing interest of multiagent system research in employing the theory of *argumentation* for the development of communication protocols. While significant progress has been made in formalising argument-based communication, (possibly *adaptive*) agent-level *argumentation strategies* as a *practical* integration of rational agent reasoning and inter-agent argumentation dialogues have received fairly little attention. In this paper we propose the use of the InFFrA framework in argument-based negotiation. This framework allows for a strategic and adaptive communication to achieve private goals within the limits of bounded rationality in open argumentation communities. The feasibility of the approach is illustrated in an agent-based web linkage scenario, showing that its performance is comparable to that of simple proposal-based negotiation while accommodating much stricter constraints regarding "what can be said" like those used in argumentation.

1 Introduction

Communication between intelligent agents is one of the cornerstones of multiagent system (MAS) technology. Most of the time, this communication is realised in terms of (1) agent communication languages (ACLs) defining the structure of messages (usually in a speech-act like format) and (2) interaction protocols specifying admissible sequences of messages and imposing constraints on the contents of these.

Recent years have seen an increasing interest in employing the theory of *argumentation* for the development of communication protocols. This interest rests on the view that *rational agents* reason and make decisions by constructing and comparing arguments for and against particular conclusions [12,10]. Hence, it is natural to view *rational interaction* as a disciplined process of argument exchange. As a consequence, significant progress has been made in formalising argument-based communication, founded in various formal theories of argumentation (e.g. [3,2,13]). One area of particular interest is *argumentation based negotiation* (ABN) [16], in which agents exchange arguments in order to reach beneficial agreements.

So far, however, fairly little attention has been paid to *argumentation strategies* as a *practical* integration of intra-agent rational reasoning and (hence, rational) inter-agent

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 122–137, 2006.

dialogue via argument exchange.[1] Even less work has been done on argumentation strategy adaptation. This is paralleled in general ACL research (see e.g. [11,6] for recent overviews of the field), where the problem of coming up with an optimal communication strategy that *obeys* a given semantics,[2] but still ensures beneficial interaction outcomes for the agents themselves, is largely unresolved.

One reason why argumentation strategies and their adaptation have received fairly little attention may be the relatively high expressiveness of argumentation protocols, which makes them both difficult to implement and hard to control during execution. In the light of the latter problem, however, the need for adaptive argumentation strategies becomes even more pressing.

In this paper, we take a first step towards the use of adaptive strategies in argument-based dialogues. For this, we conceptually follow a generic agent-centric model of strategic interaction that makes a clear distinction between agents' social behaviour on the one hand and their internal rational reasoning on the other. Rational interaction then means that agents may only engage in communication in a way that respects their alleged mental state (as suggested by their social behaviour). We practically implement a simplified form of argument-based negotiation using a particular instance of the InFFrA social reasoning framework [19]. We illustrate the feasibility of the approach in a agent-based web linkage scenario, and show that its performance is comparable to that of simple proposal-based negotiation while imposing much stricter and much more realistic constraints.

The paper contributes to the state of the art in argument-based communication in two main ways. First, it presents the first attempt to produce a highly expressive and flexible approach to adaptive communication strategies in argument-based communication in general, and argument-based negotiation in particular. Second, our practical implementation contributes to bridging the gap between global (argumentation) protocol design and rational agent design.

The remainder of this paper is structured as follows. We first introduce a generic model of strategic interaction in a prescriptive social context. In Section 3, we then lay out our approach for adaptive reasoning about communication patterns. Section 4 introduces *interest-based negotiation*, a form of argumentation-based negotiation, which will serve as an underlying argumentation model in an application scenario. Experimental results obtained in this scenario will be given in section 5. Section 6 rounds up with some conclusions.

2 A Generic Model of Strategic Interaction

As suggested in the introductory section, the perspective we adopt towards argumentation (and towards communication in general) is entirely agent-centric. The problem we are trying to solve can be stated as follows:

[1] Preliminary work has begun to investigate the outcomes of different simple strategies in argument-based communication [1,14].

[2] We mean *ostensible* adherence to a given semantics here, e.g. by acting *as if* being in a particular mental state (for mentalistic semantics [22,5]) or *as if* entering into a social commitment (for commitment-based semantics [23,9]).

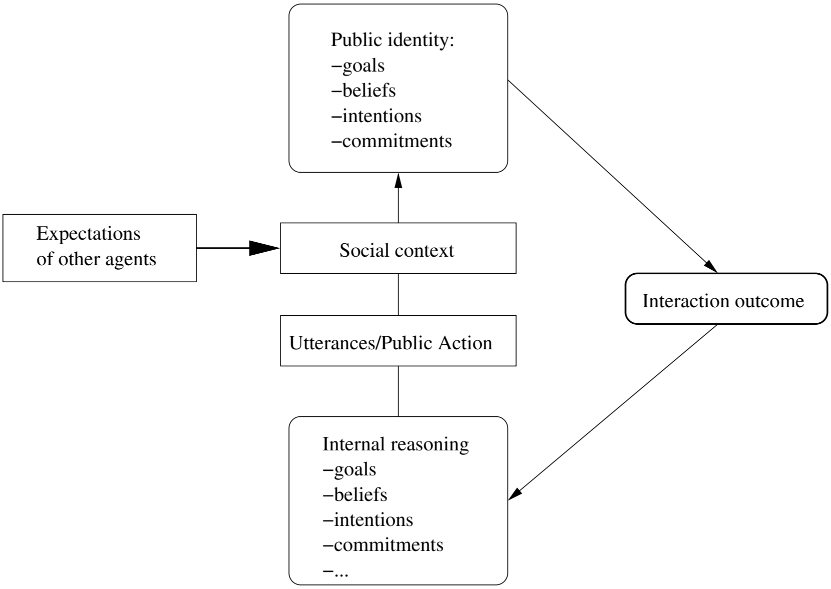


Fig. 1. Agent-centric model of strategic interaction

> *Given a set of dialogue patterns tied to constraints regarding (among other things) participants' internal structure, how can we design an agent that is capable of employing these patterns – in compliance with the constraints at hand – in order to optimise her own long-term profit?*

We should take a minute to describe what is meant by this. In the most general sense, any communication mechanism in a MAS ties the use of certain communicative patterns (protocols, single utterances or publicly observable "non-linguistic" or "physical", i.e. environment-manipulating, actions) to specific constraints, which may also concern mental states (such as beliefs, desires, and intentions [4]) particularly of those agents taking part in the communication. Since these prescriptive constraints are usually assumed to be common knowledge for all agents situated in the same *social context*, they (together with the actual communication) raise mutual *expectations* regarding these agents' behaviours. As figure 1 suggests, anything that is uttered by an agent is interpreted on the grounds of the social context and leads to the construction of a *public identity* of the communicating agent. This identity reflects (1) what the agent has publicly claimed about her internal state and (2) how this is seen to relate to her actual behaviour by other agents. However, although her utterances are generated by the agent's internal reasoning mechanism, the actual internal state may differ significantly from her public identity, and this is where the strategic aspect of communication comes into play: In contrast to the internal state of an agent, her public identity is subject to inspection by a peer and will evoke certain reactions on her peers' side and hence affect the outcome of any communicative interaction. Thus, the agent herself has an incentive to manipulate her public identity, albeit the social context confines the range of possible manipulations.

ABN is an instance of this model that allows agents to influence their public identity directly, namely by conveying information about their (alleged) internal state in order to support their proposals or claims. While an agent might put forth arguments that are not in line with her internal state, the public identity itself must be consistent so as not to reveal this difference. Otherwise, other agents might simply refuse to interact with the agent at all due to the latter's inability to interact coherently.

As an example, consider an insincere agent A who is deliberately deceiving agent B when claiming that she is trying to achieve goal G. In a framework of rational communication, A would be expected to act (and speak) in accordance with this commitment to G, e.g. by not claiming to pursue contradictory goals at the same time, by dropping the goal if it is achieved or if it becomes unachievable, etc. It is reasonable to assume that unless A says anything that contradicts these principles of rationality (which can be seen as a social context in the context of communication) or indicates through her physical actions not to be pursuing G, she can maintain her public identity towards B and keep B thinking she is in fact trying to achieve G. This is exactly what is meant by strategic compliance to a social context: to ensure others have certain expectations about our own future behaviour, we must succeed in maintaining a communicative stance that is in concordance with the social context, even if our internal state and reasoning contradicts the construed public identity.

3 Reasoning with Adaptive Communication Patterns

To develop agents that are capable of strategically dealing with the complex dialogue patterns required for argumentation, we make use of the abstract social reasoning framework InFFrA proposed in [21], and, more specifically, the formal model $\mathsf{m}^2\mathsf{InFFrA}$ suggested as a concrete instance of InFFrA in [7].

The Interaction Frames and Framing Architecture InFFrA is an abstract framework for reasoning about and learning different classes of interactions in the form of so-called *interaction frames* (henceforth called frames). Each of these frames characterises a category of interaction situations in terms of (1) *roles* held by the interacting parties and *relationships* between them, (2) *trajectories* that describe the observable surface structure of the interaction, and (3) *context* and *belief* conditions that need to hold for the respective frame to be enacted. Further, InFFrA defines *framing* as the activity of constructing, adapting and strategically applying a set of interaction frames from the point of view (and in accordance with the private goals) of a single agent. Roughly speaking, framing consists of four phases:

1. Interpreting the current interaction in terms of a *perceived frame* and matching it against the normative model of the *active frame* which determines what the interaction should look like.
2. Assessing the active frame (based on whether its conditions are currently met, whether its surface structure resembles the perceived interaction sequence, and whether it serves the agent's own goals).

3. Deciding on whether to retain the current active frame or whether to *re-frame* (i.e. to retrieve a more suitable frame from one's frame repository or to adjust an existing frame model to match the current interaction situation and the agent's current needs) on the grounds of the previous assessment phase.
4. Using the active frame to determine one's next (communicative, social) action, i.e. apply the active frame as a prescriptive model of social behaviour in the current interaction.

The m^{2}InFFrA framework turns these abstract concepts of interaction frames and framing into a concrete computational model for discrete-time, two-party, turn-taking interactions. A frame in m^{2}InFFrA consists of (1) a *trajectory*, i.e. a linear sequence of message or "physical" action patterns (possibly containing variables), (2) condition/substitution pairs that represent past enactments of the frame in terms of variable values and conditions that held at the time of the enactment, and (3) counters that keep track of (i) the frequency with which an *encounter prefix* (i.e. an initial sub-sequence of a perceived conversation) matched the frame trajectory and (ii) the frequency with which certain condition/substitution pairs appeared as instances of the frame.

3.1 Interaction Frames

To define interaction frames formally, we assume a language of speech-act like message and action patterns of the form $\texttt{perf}(A, B, X)$ or $\texttt{do}(A, Ac)$ which may contain variables or concrete values in the sender, receiver and content slots. In the case of messages (i.e., exchanged textual signals), `perf` is a performative symbol (e.g. `request`, `inform`), A and B are agent identifiers or agent variables and X is the content of the message taken from a first-order language $\mathcal{L}$.

In the case of physical actions (i.e., actions that manipulate the physical environment) with the pseudo-performative `do`, Ac is the action executed by A (a physical action has no recipient as it is assumed to be observable by any agent in the system). Both X and Ac may contain non-logical "substitution" variables used for generalisation purposes (as opposed to logical "content" variables used by agents to indicate quantification or to ask for a valid binding).

Interaction frames are then defined as tuples $F = (T, \Theta, C, h, h_\Theta)$, where

- $T = \langle p_1, p_2, \ldots, p_n \rangle$ is a sequence of message and action patterns p_i, the *trajectory*,
- $\Theta = \langle \vartheta_1, \ldots, \vartheta_m \rangle$ is an ordered list of *variable substitutions*,
- $C = \langle c_1, \ldots, c_m \rangle$ is an ordered list of *condition sets*, such that $c_j \in 2^{\mathcal{L}}$ is the condition set relevant under substitution ϑ_j,
- $h \in \mathbb{N}^{|T|}$ is a *trajectory occurrence counter* list counting the occurrence of each prefix of the trajectory T in previous conversations, and
- $h_\Theta \in \mathbb{N}^{|\Theta|}$ is a *substitution occurrence counter* list counting the occurrence of each member of the substitution list Θ in previous conversations.

While the trajectory $T(F)$ models the surface structure of message sequences that are admissible according to frame F, each element of $\Theta(F)$ resembles a past binding of the variables in $T(F)$, and the corresponding element of $C(F)$ lists the conditions required for or precipitated by the execution of F in this particular case. $h(F)$ finally indicates how often F has been executed completely or just in part, $h_\Theta(F)$ is used to avoid duplicates in $\Theta(F)$ and $C(F)$. What hence distinguishes interaction frames from the methods commonly used for the specification of ACL and protocol semantics is that they allow for an explicit representation of *experience* regarding their practical use.

To illustrate these definitions, consider the following example, in which we use an abbreviated notation to capture all elements of the definition more concisely:

$$\begin{aligned} F = \; \Big\langle & \overbrace{\langle \xrightarrow{5} \texttt{request}(A,B,X) \xrightarrow{3} \texttt{do}(B,X)\rangle}^{T}, \\ & \langle \overbrace{\{can(B,X)\}}^{\Theta_1}, \overbrace{\{can(B,\texttt{pay}(S)\}}^{\Theta_2} \rangle \\ & \langle \overbrace{\xrightarrow{2} \langle [A/\texttt{a}],[B/\texttt{b}],[X/\texttt{pay}(\$100)]\rangle,}^{C_1} \\ & \overbrace{\xrightarrow{1} \langle [A/\texttt{b}],[B/\texttt{a}],[X/\texttt{pay}(S)]\rangle}^{C_2} \rangle \Big\rangle \end{aligned}$$

As for the individual elements of F, the *trajectory* T captures the following interaction experience: A has asked B five times to perform (physical) action X, B actually did so in three of these instances.[3] Knowledge about the remaining two cases would typically be stored in a different frame. The substitutions Θ and conditions C summarise the following observations: In two of the successful instances (Θ_1/C_1), it was a who asked and b who heeded the request, and the action was to pay \$100. In both cases, $can(\texttt{b}, \texttt{pay}(\$100))$ held true (C_i always corresponds to Θ_i in a frame). In the third case, roles were swapped between a and b and the amount S remains unspecified (which does not mean that it did not have a concrete value, but simply that this was abstracted away in the frame).

Thus, m^{2}InFFrA frames facilitate the description of (observations about) dialogue sequences by means of generalised message (and action) patterns together with past variable values and context conditions. At the same time, they can be used as a concrete representation for the abstract social context mentioned above, combining behavioural expectations and context conditions in the most general way.

3.2 Frame Semantics

As for the semantics of frames, these are defined in terms of a probabilistic model over the possible *continuations* of a dialogue, i.e. current frame knowledge induces a probability distribution over possible conclusions to a dialogue given a prefix sequence of what has been observed in a dialogue at a certain point in time. This is done using a domain-dependent real-valued similarity measure σ on message patterns (and sequences thereof). σ is defined using a distance metric between messages[4] and

[3] do is used as a special performative to indicate execution of physical actions.

[4] See [8] for details on this metric.

extended into a similarity $\sigma(\vartheta, F)$ between a substitution ϑ and an entire frame F with trajectory $T(F)$ and substitutions $\Theta(F)$ by summing over individual similarities $\sigma(T(F)\vartheta, T\Theta_i(F))$ between the message pattern sequence induced by ϑ and the past cases stored in $\Theta(F)$. Moreover, $\Theta_i(F)$ is only considered if the corresponding condition set $C_i(F)$ is currently satisfied.[5] This means that certain valuations for variables in the message "templates" of the trajectory that have been observed in the past will only be considered if the conditions under which they were observed hold in the current state.

Given a frame repository $\mathcal{F} = \{F_1, \ldots F_n\}$ representing the agent's interaction experience, and a (possibly empty) sequence w of messages perceived in the current conversation, the probability of encounter prefix w being concluded with w' computes as

$$P(w'|w) = \sum_{F \in \mathcal{F}, ww' = T(F)\vartheta} P(\vartheta|F, w)P(F|w),$$

i.e. the probability that some F is enacted under a specific substitution ϑ such that ww' equals the trajectory of F under ϑ. To compute the probabilities on the right-hand side of this equation, we assume that $P(\vartheta|F, w) \propto \sigma(\vartheta, F)$ in the sense that the likelihood of any substitution is proportional to its similarity to a frame as compared to that of any other substitution still possible. The probability $P(F|w)$ is computed by looking at the occurrence counter value corresponding to the last element of $T(F)$ (i.e. to $T(F)$ as a whole).

3.3 Decision Making and Frame Adaptation

Based on this probabilistic semantics, [19] defines a two-layer decision-making and learning process: at the (lower) action level, agents use utility estimates $u(w, KB)$ (which are obtained, for example, by computing the utility of physical (do) actions that occur along a dialogue sequence w under current knowledge KB and assigning a small communication cost to each "non-physical" message) to maximise the expected utility *within* the activated frame (i.e. among all substitutions that this frame still permits). This involves adversarial search in the space of variable substitutions that the agent and her peer may apply in their respective part(s) of the conversation (since values for some of these variables can be chosen by the agent and for others this is done by her peer).

At the (upper) *framing* level, which is concerned with choosing a frame to activate from a given frame repository $\mathcal{F}$, agents use a variant of hierarchical Q-learning (based on the *options* framework proposed in [15]) to learn optimal re-framing strategies for changing frames during an encounter if (1) the current frame trajectory no longer matches the perceived encounter message sequence, (2) frame conditions no longer apply, or (3) the frame no longer seems to offer positive utility under the optimal substitution.

The intuition behind this layered approach is that frames provide decision-making blocks for communication behaviour that help the agent distinguish between different

[5] This assumes the agent maintains some kind of knowledge base KB and can verify $KB \models \varphi$ for any φ entailed by KB.

communication situations, e.g. different phases in a dialogue (e.g. "preference elicitation" and "proposal exchange" in automated negotiation contexts). Once an appropriate frame has been identified for the current situation (which also depends on the context conditions contained in that frame) the alternatives provided by other frames can safely be discarded, at least until the frame has been successfully executed or some problem arises and *re-framing* occurs. While no re-framing occurs, the agent activates the lower decision-making layer to make optimal decisions regarding the degrees of freedom that are provided by the currently active frame (in the form of unbound variables). Since this search space is manageable, we conduct exhaustive (adversarial) search based on maximum expected utility of all ground continuation sequences the frame caters for. This allows for communicative decision making under *bounded rationality* conditions.

The full $\mathsf{m}^2\mathsf{InFFrA}$ architecture contains several additional components:

- *A mechanism for frame adaptation generalisation from experience.* Since agents may deviate from pre-defined frames, frame adaptation cannot be restricted to a mere update/extension of counter values and condition/substitution sets. To allow for the adaptation of frames from actual interaction experience, [8] extends the aforementioned distance metric on message sequences to frames and interprets frames as clusters in the space of possible conversations. *Cluster validation* techniques are then used to decide whether (and how) new observations should be merged into existing frames or whether they should be used to create a new frame.
- *Heuristics for making decisions about deviating from existing frames.* Often, agents would prefer to deviate from the currently executed frame because it does not seem desirable under changed environmental circumstances (e.g. by not executing costly physical actions that are part of the trajectory). However, this would jeopardize the long-term stability of the whole system of frames and trust in their use and therefore we have developed heuristics to facilitate an explicit trade-off between local desirability and global predictability of communication processes [20].
- *Methods for deriving encounter state abstractions.* At the "upper" frame selection decision-making level, agents must base their choices on the current "communication state", which has to be modelled in a different way from the general state of affairs (which is essentially described by all facts in the agent's knowledge base KB). In [19], we describe methods for deriving such abstractions of the general state of the world by focusing on those aspects of it that are relevant to the current dialogue, such as the role of the agent in it, the subject of the dialogue and its projected effect(s).

For lack of space, we will omit these details of the architecture here. For our purposes, it shall suffice to note that $\mathsf{m}^2\mathsf{InFFrA}$ provides a framework for decision-theoretic (boundedly) rational selection and long-term adaptation of dialogue patterns in the form of simple interaction frames. This is achieved by providing agents with an initial set of admissible patterns, to which they will add their experiences over time. Based on similarity considerations and long-term accumulation of feedback regarding the usefulness of different frames in different interaction situations, they can optimise their frame and action choices. In the following, we will use this architecture for learning and decision-making in a complex social context, namely that of interest-based negotiation.

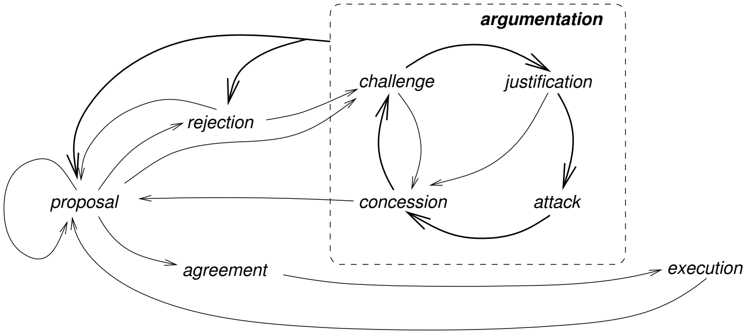


Fig. 2. Basic control flow of interest-based negotiation

4 Interest-Based Negotiation

In contrast to *proposal-based* negotiation, in which agents merely exchange proposals (such as contracts in contract nets, deals in bargaining or goods and prices in auctions), *argumentation-based* negotiation [16] allows agents to exchange information about their internal state in order to convince the other that a particular course of joint action will be mutually beneficial. *Interest-based negotiation* (IBN) [17] is a particular ABN framework which allows agents to argue over each others' beliefs, goals, and the means for achieving these goals. In this paper, we are concerned with a simplified variant of IBN in which an agents' proposal may be (1) challenged by asking for reasons (in terms of the agent's beliefs, goals, etc.) that lead to her negotiation stance and (2) justified by the agent, whereupon the challenging agent may (3) attack this justification until finally the attack, if successful, leads to a (4) concession that brings the agents closer to an agreement. Figure 2 illustrates the basic control flow of the dialogues we model here. The original IBN framework described in [17] enables more flexible dialogues (e.g. involving shifts in focus during the dialogue). As a starting point and to make the simulation viable, however, we consider only a subset of these possible dialogues and explore strategy learning within this subset.[6]

The process of IBN can be seen as traversal of a so-called *goal graph* that facilitates the representation of goal hierarchies, preferences and justifications. Each node in a goal graph represents a fact or a goal, and directed links between goal nodes can be used to represent goal hierarchies. Furthermore, a link (viz set of links) leading from a fact (viz set of facts) to a goal node and labelled with an action identifier denotes that execution of this action requires the respective fact to be true, and contributes to the respective goal. In terms of the model presented in section 2, a goal graph constructed from the arguments put forward by a specific agent can serve as a representation of this agent's public identity. An example of a goal graph for a particular domain will be given in section 5.

What makes IBN attractive for the study of argumentation strategy learning and the reason why IBN lends itself well to an implementation in $\mathsf{m^2InFFrA}$ is that it provides

[6] In the remainder of the paper, we will use the term IBN to refer to this simpler version of interest-based negotiation.

Table 1. Example frames for single-shot IBN

$$\Big\langle \big\langle \rightarrow \texttt{request}(A,B,X) \rightarrow \texttt{reject}(B,A,X) \rightarrow \texttt{ask-reason}(A,B,X) \rightarrow \texttt{inf-problem}(B,A,P) \rightarrow \texttt{concede}(A,B,P)\big\rangle, \big\langle\{\mathit{problem}(P,X)\}\big\rangle, \big\langle\langle\rangle\big\rangle\Big\rangle$$

$$\Big\langle \big\langle \rightarrow \texttt{request}(A,B,X) \rightarrow \texttt{ask-reason}(B,A,X) \rightarrow \texttt{inf-goal}(A,B,G) \rightarrow \texttt{att-means}(B,A,Y) \rightarrow \texttt{concede}(A,B,Y) \rightarrow \texttt{do}(B,Y)\big\rangle, \big\langle\{\mathit{goal}(A,G), \mathit{achieves}(X,G), \mathit{achieves}(Y,G), X \neq Y\}\big\rangle, \big\langle\langle\rangle\big\rangle\Big\rangle$$

a rich set of social rules with which agents have to comply when engaging in rational argumentation. The most prominent of these are:

1. No proposal can be considered viable if it cannot be implemented under current circumstances, i.e. if its environmental preconditions are not met. If an agent is informed (believably) that her proposal rests on false assumptions, she must withdraw it.
2. A proposal has to be dropped if it can be shown that its effects have already been achieved or that they are unachievable.
3. No proposal is acceptable that violates a higher-level goal even though it achieves some lower-level goal. In fact, to make things more difficult we will require that a proposal that jeopardises *any* goal will be considered unacceptable in our experiments.
4. No alternative to a proposal can be rejected once the fact has been accepted that it will achieve the alleged goal (i.e. if it achieves the same thing, there is no reason to reject an alternative).

Note that in this list, "goals" always refer to the agent's *own* goals, i.e. we do not assume any "collective rationality" that would force the agent to justify her stances with respect to a global set of goals. We rather assume that the public identity of the agent is described by a goal and belief structure that the *agent* is supposed to have, and in communication she has to act in accordance with this purported goal structure.

Quite interestingly, while these rules are based on principles of agent-level rationality (some of them in fact reflect fundamental elements of BDI theory [4]), in an argumentation scenario they constitute society-level rules of communicative behaviour: Any agent who violates them would no longer be treated as rational by others and might be excluded from the society altogether (simply for lacking the ability to participate in reasonable communication).

In the context of m^{2}InFFrA, these rules can be used directly to define argumentation frames. Consider the two frames for single-shot[7] IBN quoted from [19] and shown in table 1, which implement rules 1 and 4, respectively. In the first frame, B justifies her

[7] I.e., involving only one iteration of the challenge-justification-attack-concession loop shown in figure 2. For lack of space, the frames developed in [19] for iterative IBN are omitted here.

refusal to perform the proposed action X by pointing to a problem P that inhibits execution of X. In the second frame, B attacks A's justification for action X (namely a goal G achieved by it) with an alternative action Y that achieves G as well. The logical predicates $problem$, $goal$ and $achieves$ in these examples (their meaning should be obvious from the context) refer to knowledge states of the individual agents. While it may be fairly easy for both agents to check specific instances of $problem$ and $achieves$ (e.g. by "inspecting" the environment), this is certainly not the case for the $goal$ predicate. However, this statement still has to be consistent with the public identity of agent A as given by her past (and future) statements. As a result, A cannot attack the alternative means Y for achieving G, independent of the fact if she really holds G as a goal.[8]

To allow for the exchange of multiple arguments, [19] further defines six frames for iterative IBN, corresponding to a successful proposal, challenged proposal, and rejected proposal (i.e. edges leading out of the proposal node of figure 2) and to successful challenge, successful justification, and successful attack (i.e. edges leading into the concession node). Using these frames in practice (or, more precisely, using an indefinitely long sequence of frames in a single encounter) requires a more complex control flow than that currently possible in m^{2}InFFrA (e.g. storing the state of a particular argument or proposal when a shift in focus occurs). While beyond the scope of this paper, this certainly is on our research agenda in order to increase the expressiveness and flexibility of our implementation.

The overall workings of a society of IBN-m^{2}InFFrA agents in the experiments that we report on below are as follows:

- We equip all agents in the society with a set of (identical) interaction frames that enable them to conduct an IBN process as depicted in 2). Initially, all counters in these frames are set to 0 and substitution sets are empty.
- We construct a goal graph which can be inspected by any agent and that reflects the goal structure of a rational agent in the domain. In principle, agents could describe their internal (alleged) goal structures to each other through discussions from scratch, but we assume a commonly known goal structure to simplify things because we are only interested in rational argumentation *given* some publicised goal structure.
- Although this is not required in the general m^{2}InFFrA architecture, we force agents to adhere to these frames, i.e. once activated, agents cannot deviate from them unless in ways permitted by the inital . In other words, we require agents to obey the overall communicative regime, which will in this case force them to concede to anything that follows from their assumed goal structure and beliefs. As a consequence, the strategic choices of agents are restricted to *which* of the currently matching frames to use *when* and with which concrete values for unbound variables in that frame in any given step.
- As interaction unfolds, agents will adapt their frame repositories according to observations and attempt to optimise their long-term strategy using the hierarchial

[8] For reasons of simplicity, we suffice with this, somewhat naive, approach for handling the complex notion of *commitment* in dialogue [25]. An elaborate account of commitment management is beyond the scope of this paper.

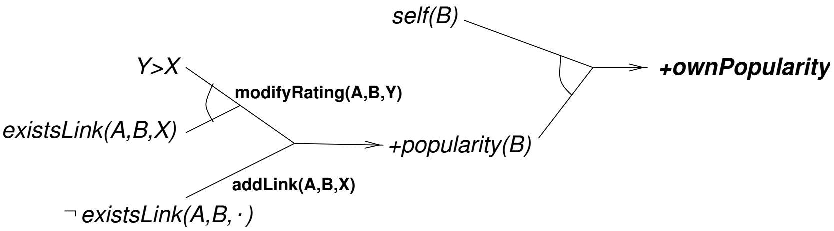


Fig. 3. Part of the LIESON goal graph for the (sub)goal of increasing one's own popularity (by either obtaining an in-link with arbitrary rating value or increasing an having an existing link's rating value X be increased to Y), which itself contributes to the (super)goal of increasing one's score

learning and optimisation process described for m^{2}InFFrA above. They will be solely judged by the "physical" utilities they obtain from the environment, i.e. no genuine, immediate gain can be derived from communication itself (other than a small communicative (negative) cost).

5 Experimental Results

As a proof of concept, interest-based negotiation using interaction frames has been implemented and tested in the multiagent-based link exchange system LIESON. In this system, agents representing Web sites engage in communication to negotiate over mutual linkage with the end of increasing the popularity of one's own site and that of other preferred sites.

Available physical actions in this domain are the addition and deletion of numerically rated links originating from one's own site and the modification of these ratings (where the probability of attracting traffic through a link depends on the rating value). Agent performance is computed based on the flow through the link network as well as on private ratings the agents hold towards each other. It is worth noticing that these private ratings also introduce a form of "social standing", since linkage decisions by higher rated agents have a greater impact on individual as well as overall performance. Figure 3 shows a part of the goal graph for LIESON.

Technically, LIESON agents consist of a non-social BDI [18] reasoning kernel that projects future link network configurations and prioritises goals according to utility considerations. If these goals involve actions that have to be executed by other agents, the m^{2}InFFrA component starts a framing process which runs until the goal of communication has been achieved or no adequate frame can be found. Note that the goal-prioritising internal reasoning mechanism need not generate intentions that are in keeping with the goal graphs talked about in conversations with other agents. This is exactly what is meant by strategically exploiting the possibilities of public identity management while pursuing one's private agenda.

We report on two different sets of experiments in order to compare the performance of simple proposal-based negotiation (PBN) to that of (single-shot) IBN. PBN has been implemented by supplying agents with a set of frames that allow for requesting action

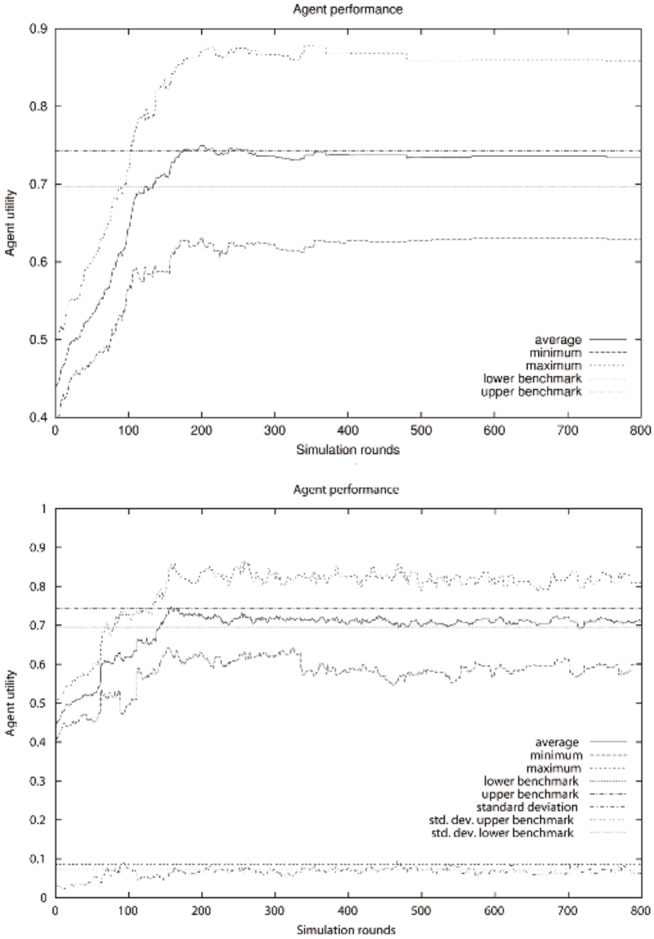


Fig. 4. Performance plots for proposal-based (above) vs. interest-based negotiation (below)

execution from another agent, proposing alternative actions, or proposing actions the other has to perform in return for one's cooperation. In contrast to the IBN case, agents are free to perform or not perform these actions without giving a reason. Single-shot IBN is realised using a set of frames, one for each path in the graph of figure 2 (two examples for these were given in the previous section, for exact definitions of the remaining frames cf. [19]). As compared to proposal-based negotiation, IBN enforces a much stricter *communication regime* by requiring agents to justify their stance, to accept any alternative suggested for the same goal, to abandon any proposal that threatens at least one goal, etc. In particular, this also implies that agents cannot simply reject a proposal because it does not seem desirable in terms of utility.

Figure 4 shows the average agent performance (in a society of ten agents) as well as the individual performances of the best and worst agent for PBN (above) and IBN (below), both of them averaged over 50 independent runs. The constant lines depicted in the plots correspond to benchmark values that are relevant in the linkage domain: the lower benchmark corresponds to the average utility that can be achieved if every agent "honestly" expresses her opinion towards any other agent by laying a link that is weighted with her actual rating for the target site, while the upper benchmark is reached if every agent lays "politically correct" links to all other agents by not laying any links with negative ratings (i.e. concealing any critical views of other sites). Note that to start off with, agents know nothing about these benchmarks and link configurations that will yield high utility scores to them. In particular, laying more and more links in an honest way is only slightly dominated by the strategically superior, politically correct linkage pattern and it is quite impressive that agents achieve a performance close to the upper benchmark.

The significance of the results shown in figure 4 is that the agents manage to attain (and maintain) a reasonable level of long-term utility even under these stricter – and much more realistic – circumstances (albeit with bigger fluctuations indicating frequent "loss of an argument"). This illustrates nicely that m^{2}InFFrA is capable of combining decision-theoretic learning with complex knowledge-based reasoning about constraint-governed conversation patterns. The ability to record experiences with certain communication patterns (by extending the pre-specified negotiation frames with new substitutions and situation-dependent conditions) and to reinforce their use depending on the environmental feedback obtained while using them in a particular interaction allows agents to adapt not only to a set of communication patterns but in fact to the (evolving) *communication practice* of a MAS.

The results also suggest that IBN has an *equilibratory* effect on the social outcome since the utility difference between most and least successful agents is somewhat smaller than is the case for proposal-based negotiation. Quite naturally, the requirement to "give reasons" (and hence to act rationally in accordance with public identity) seems to reduce the impact of "having more power". Indeed, a closer look at individual interactions reveals that agents are capable of "winning an argument" independent of their power and that of their peer.

6 Conclusions

In this paper, we have presented a practical, adaptive approach to argumentation for artificial agents. Starting with a brief discussion of the general issue of strategic interaction, we have argued for a separation between *internal reasoning* and *social behaviour* to allow for the combination of the decision-theoretic design of a rational, self-interested agent with the prescriptive, society-level constraints entailed by typical argumentation protocols.

We have introduced the abstract architecture InFFrA and the notions of *interaction frames* as a representation of a class of interaction in terms of surface structure and contextual conditions and of *framing* as the process of strategically applying a set of interaction frames and adapting them from experience.

We have further presented a simplified yet flexible version of interest based negotiation, which allows for rejecting, challenging, justifying and attacking arguments in the form of agents' mental states. This version has then been implemented using a particular instance of InFFrA for two-party, turn-taking conversations. The feasibility of the approach has been illustrated in a agent-based web linkage scenario, and its performance has been shown to be comparable to that of simple proposal-based negotiation while imposing much stricter and much more realistic constraints.

To our knowledge, the implementation of IBN frames in m^2InFFrA constitutes both the first practical approach and the first application of machine learning methods to argumentation-based negotiation in MASs. This example illustrates that the combination of logical constraints (that can be used to describe knowledge-level or social-level communication semantics) and probabilistic models of communication processes (which allow for an application of decision-theoretic learning and optimisation methods) make m^2InFFrA a prime candidate for achieving rational agent behaviour in other, similarly complex communication contexts that are defined by modern ACLs and interaction protocols.

In the future, we would like to extend frame representations to enable more complex communication constraints and capture protocol information beyond simple turn-taking message sequences, in particular by allowing cycles, branching and multi-party dialogues. Also, we are interested in "context mining" for frame conditions, i.e. the automated discovery of those aspects in the context that are responsible for the success (or failure) of a frame. Finally, as frames suggest the combination of symbolic communication and constraints with it would be interesting to integrate interaction frames with existing *relational reinforcement learning* [24] methods.

References

1. L. Amgoud and N. Maudet. Strategical considerations for argumentative agents (preliminary report). In S. Benferhat and E. Giunchiglia, editors, *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR 2002): Special session on Argument, Dialogue and Decision*, pages 399–407, 2002.
2. L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In W. Horn, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 338–342, Amsterdam, Netherlands, 2000. IOS Press.
3. K. Atkinson, T. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument over proposals for action. In Iyad Rahwan, Pavlos Moraitis, and Chris Reed, editors, *Proceedings of the First International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'04)*, volume 3366 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2005.
4. M. E. Bratman. *Intentions, Plans and Practical Reason.* Harvard University Press, Cambridge, MA, 1987.
5. P. R. Cohen and H. J. Levesque. Communicative actions for artificial agents. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS)*, pages 65–72, 1995.
6. V. Dignum. *A model for organizational interaction: based on agents, founded in logic.* PhD thesis, Utrecht University, The Netherlands, 2004.
7. F. Fischer and M. Rovatsos. Reasoning about communication: A practical approach based on empirical semantics. In *Proceedings of the 8th International Workshop on Cooperative Information Agents (CIA)*, volume 3191 of *Lecture Notes in Artificial Intelligence*, Erfurt, Germany, 2004. Springer-Verlag.

8. F. Fischer, M. Rovatsos, and G. Weiß. Acquiring and adapting probabilistic models of agent conversation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Utrecht, The Netherlands, 2005.
9. N. Fornara and M. Colombetti. Operational specification of a commitment-based agent communication language. In Maria Gini, Toru Ishida, Cristiano Castelfranchi, and W. Lewis Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 536–542, Bologna, Italy, 2002. ACM Press.
10. A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 883–890, Melbourne, Australia, 2003.
11. M. T. Kone, A. Shimazu, and T. Nakajima. The state of the art in agent communication languages. *Knowledge and Information Systems*, 2:259–284, 2000.
12. P. Krause, S. Ambler, M. Elvang-Gøransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995.
13. P. McBurney. *Rational Interaction*. PhD thesis, University of Liverpool, 2002.
14. S. Parsons, M. J. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.
15. D. Precup. *Temporal Abstraction in Reinforcement Learning*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, 2000.
16. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4), 2003.
17. I. Rahwan. *Interest-based Negotiation in Multi-Agent Systems*. PhD thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia, 2004.
18. A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In W. Swartout C. Rich and B. Nebel, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R)*, pages 439–449, 1992.
19. M. Rovatsos. *Computational Interaction Frames*. PhD thesis, Department of Informatics, Technical University of Munich, 2004.
20. M. Rovatsos, M. Nickles, and G. Weiss. Interaction is meaning: A new model for communication in open systems. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003.
21. M. Rovatsos, G. Weiß, and M. Wolf. An Approach to the Analysis and Design of Multiagent Systems based on Interaction Frames. In Maria Gini, Toru Ishida, Cristiano Castelfranchi, and W. Lewis Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Bologna, Italy, 2002. ACM Press.
22. M. P. Singh. A semantics for speech acts. *Annals of Mathematics and Artificial Intelligence*, 8(1–2):47–71, 1993.
23. M. P. Singh. A social semantics for agent communication languages. In *Proceedings of the IJCAI Workshop on Agent Communication Languages*, 2000.
24. P. Tadepalli, R. Givan, and K. Driessens. Relational Reinforcement Learning: An Overview. In *Proceedings of the Workshop on Relational Reinforcement Learning*, Banff, Alberta, Canada, 2004.
25. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany NY, USA, 1995.

A Protocol for Arguing About Rejections in Negotiation

Jelle van Veenen[1] and Henry Prakken[2,3]

[1] Faculty of Law, Tilburg University, The Netherlands
j.vanveenen@uvt.nl
[2] Department of Information and Computing Sciences,
Utrecht University, The Netherlands
henry@cs.uu.nl
[3] Centre for Law & ICT, Faculty of Law, University of Groningen, The Netherlands

Abstract. One form of argument-based negotiation is when agents argue about why an offer was rejected. If an agent can state a reason for a rejection of an offer, the negotiation process may become more efficient since the other agent can take this reason into account when making new offers. Also, if a reason for rejection can be disputed, the negotiation process may be of higher quality since flawed reasons may be revised as a result. This paper presents a formal protocol for negotiation in which reasons can be asked and given for rejections and in which agents can try to persuade each other that a reason is or is not acceptable. The protocol is modelled as a persuasion dialogue game embedded in a negotiation protocol. It has a social semantics since the protocol does not refer to the internal state of negotiating agents.

1 Introduction

Recently argumentation-based approaches to negotiation have become popular (see [1] for an overview and motivation). The idea is that if negotiating agents exchange reasons for their proposals and rejections, the negotiation process may become more efficient and the negotiation outcome may be of higher quality. This paper especially focuses on reasons given for rejections of proposals. If an agent explains why he rejects a proposal, the other agent knows which of her future proposals will certainly be rejected so she will not waste effort at such proposals. Thus efficiency is promoted. In such exchanges, reasons are not only exchanged, they can also become the subject of debate. Suppose a car seller offers a Peugeot to the customer but the customer rejects the offer on the grounds that French cars are not safe enough. The car seller might then try to persuade the customer that he is mistaken about the safety of French cars. If she succeeds in persuading the customer that he was wrong, she can still offer her Peugeot. Thus the quality of the negotiation is promoted, since the buyer has revised his preferences to bring them in agreement with reality.

This example illustrates that a negotiation dialogue (where the aim is to reach a deal) sometimes contains an embedded persuasion dialogue (where the aim is

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 138–153, 2006.

to resolve a conflict of opinion). The aim of this paper is to formulate a protocol for negotiation with embedded persuasion dialogues about the reasons for rejecting a proposal. The key idea is that the propositional commitments incurred by the agents in the embedded persuasion dialogue constrain their behaviour in the surrounding negotiation dialogue. In agreement with the current trend [2] we intend the combined protocol to have a social semantics. For this reason we will completely abstract from the internal design of the communicating agents; in particular, the protocol will only refer to the agents' publicly observable behaviour within a dialogue. According to [2] a social semantics is desirable for agent interaction protocols since if a protocol refers to an agent's mental state, there is no gurantee that an outside observer can verify whether the agent complies with a protocl.

The main novelty of the present research lies in the fact that current protocols for argument-based negotiation only allow arguments supporting proposals. One exception is [3], which also allows arguments about rejections. However, their protocol does not have a social semantics, since whether an agent is allowed to assert a claim or an argument partly depends on their internal mental state. The protocol has some other limitations, which will be discussed in Section 5.

Our proposal will be stated in a dialogue game form. It will combine a negotiation protocol and language of [4] with a persuasion protocol based on [5], which adapts and extends work of [6]. In the following sections we will first introduce these two systems and sketch the underlying argumentation logic that we will use. Then we will present our combined protocol, investigate some of its formal properties and illustrate it with an example.

2 The Building Blocks

In this section we present the negotiation and persuasion system that we aim to combine. Both systems are formulated as a dialogue game. Dialogue games formulate principles for coherent dialogue, and coherence depends on the goal of a dialogue. The goal of negotiation dialogues is to reach agreement on the division of scarce resources [1] and the goal of persuasion dialogues is to resolve a conflict of opinion [7]. Formal dialogue games have a *topic language* L_t with a logic $\mathcal{L}$, and a *communication language* L_c with a *protocol* P. The protocol specifies the allowed moves at each point in a dialogue. A dialogue system also has *effect rules*, which specify the effects of utterances on the participants' commitments, and *outcome rules*, defining the outcome of a dialogue.

2.1 A Language and Protocol for Multi-attribute Negotiation

The negotiation system we will use is that of Wooldridge and Parsons [4]. The **negotiation topic language** L_t^n of this system assumes that in a negotiation agents try to reach agreement over the values of a finite set $V = \{v_1, ..., v_m\}$ of *negotiation issues*. Each issue v can be assigned at most one value from a range $C(v)$ of values. An *outcome* of a negotiation is an assignment of values to a subset of V. A proposal is expressed in a subset of the language of first-order logic as a conjunction of expressions of the form vRc, where $v \in V$ and $c \in C(v)$ or $c =?$, (where ? technically is a free variable, capturing that the issue has not been assigned a value) and R denotes one of the relations $=, <, >, \leq$ or $\geq$.

The **negotiation communication language** L_c^n can be used to talk about proposals. The left column of table 1 shows the speech acts that agents can perform and the right column their possible replies. The formulas φ and φ' are elements of L_t^n. *Request(φ)* is a request for an offer. Here φ typically is wholly or partially uninstantiated (i.e., it may contains occurrences of ?): the speech act $request(price =? \wedge warranty = 12)$ can be read as "What is the price if I want a 12 months warranty?". The speech act *offer*(φ) makes a fully instantiated proposal φ, and with *accept*(φ) an agent accepts an offer φ made by another agent. With *reject*(φ) such an offer is rejected. With *withdraw* an agent withdraws from the negotiation.

We next outline the **negotiation protocol** of [4] for this language, with notation slightly adapted to our purposes. A negotiation takes place between two agents, one of whom starts with either an *offer* or a *request*. The agents then take turns after each utterance, selecting their replies from Table 1. As the table indicates, a negotiation terminates when an agent accepts an offer or withdraws from the negotiation. Finally, moves may not be repeated by the same player.

Table 1. Speech acts and replies in L_c^n

Acts	Replies:
request(φ)	*offer(φ')*
offer(φ)	*offer(φ')* or *accept(φ)* or *reject(φ)* or *withdraw*
reject(φ)	*offer(φ')* or *withdraw*
accept(φ)	end of negotiation
withdraw	end of negotiation

($\varphi \neq \varphi'$)

To ensure that the offers exchanged during a negotiation and its outcome are related to an initial request, we add the following rule to the protocol of [4]:

- If *request*(φ) is the initial request of a dialogue then for any move *offer*(ψ) in the dialogue:
  - ψ is logically consistent with φ; and
  - ψ contains at least the same issues as φ.

Since issues have at most one value, this rule implies that an instantiated part of a request cannot be changed by an offer (but the offer may contain more issues than the request). Therefore:

Proposition 1. *If a negotiation that starts with a request terminates with acceptance of an offer, that offer is consistent with and fully instantiates the request.*

We illustrate the system with an example in which two agents, Paul (P) and Olga (O), negotiate over the sale of a car. The dialogue starts when Paul requests to buy a car, and shows that he is interested in the brand and the price.
P_1: $request(brand = ? \wedge price = ?)$
O_2: $offer(brand = peugeot \wedge price = 10000)$
P_3: $reject\ (brand = peugeot \wedge price = 10000)$
(Olga has offered a Peugeot for 10000, but Paul has rejected the offer. Olga makes him another offer.)
O_4: $offer(brand = renault \wedge price = 8000 \wedge stereo = yes)$
P_5: $reject(brand = renault \wedge price = 8000 \wedge stereo = yes)$
O_6: $offer(brand = audi \wedge price = 10000)$
P_7: $accept(brand = audi \wedge price = 10000)$
(Olga offers a Renault with stereo for 8000. Paul again rejects after which Olga offers a non-French car for 10000. Paul accepts and the dialogue terminates. Move O_4 illustrates that an offer may introduce additional issues, for instance, to make an offer more attractive or to make a trade-off possible.)

2.2 The Underlying Argumentation Logic

We next present the logical elements assumed by our persuasion protocol, i.e., the persuasion topic language L_t^p and its logic $\mathcal{L}$. In doing so we will abstract from details of the language and inference rules wherever possible, to allow for different instantiations of the logic and language. Thus we in fact specify a set of constraints on L_t^p and $\mathcal{L}$ assumed by our persuasion dialogue system.

Much other work on argument-based dialogue, e.g. [3, 8], regards arguments as classical proofs from consistent sets of propositional formulas and allows classical inconsistency of the premises of two arguments as the only source of attack on arguments. We argue that the present application requires a richer language and notions of argument and attack. The topic language L_t^p must include a suitable subset of first-order predicate logic, to express arguments about values for negotiation issues. Since in persuasion dialogues arguments are often attacked by counterarguments, the logic $\mathcal{L}$ must be a logic for defeasible argumentation, or 'argumentation system' for short (cf. [9]). We want our system to be an instance of the well-understood abstract framework of [10], in particular of his grounded semantics (also used by e.g. [11, 12]), since this semantics can be easily incorporated into a persuasion dialogue game. Since the arguments exchanged in persuasion dialogues are often constructed stepwise during a dialogue in reply to challenges of the premises, the argumentation system must allow for a tree structure of arguments, where inference rules are chained into trees. As for notation, $prem(A)$ and $conc(A)$ denote the premises and conclusion of argument A, i.e. the leaves and root of the tree structure. Furthermore, since arguments exchanged in persuasion dialogues are often based on defeasible argumentation schemes (such as schemes for practical reasoning or default reasoning), the rules for constructing arguments must include defeasible as well as deductive inference rules. Each defeasible inference rule comes with one or more *undercutters*, which specify the circumstances under which the inference rule cannot be applied. Accordingly, a defeasible argument can be defeated in two ways. It can be *rebut* with an argument for the opposite conclusion, while it can be *undercut* with an argument why an inference rule does not apply in the given circumstances. To be successful, an attack should be of a certain strength. In the present paper, we will not discuss issues of strength and therefore implicitly assume a given measure of relative strength between arguments. Also, since our persuasion dialogue system is intended for any underlying logic satisfying the above constraints, we will not further specify the defeasible inference rules here but rather introduce them semiformally when discussing our examples. For technical details the reader is referred to e.g. [13] and [9].

Given a set of arguments and a binary defeat relation defined over it, an argumentation system classifies the arguments into justified, defensible and overruled arguments. Our persuasion system presupposes a game-theoretic formulation of Dung's grounded semantics [14, 11]. The proponent and opponent of a certain argument play a game where proponent starts with an argument he wants to defend and then both players take turns, defeating the preceding argument with a counterargument. A player wins if the other player has run out of moves. Now an argument is justified if proponent has a winning strategy in a game starting with the argument; and a proposition is justified if it is the conclusion of a justified argument. This game can be optimised in several ways (see e.g. [11]) but in order to focus on the essence we leave them undiscussed here.

2.3 A Dialogue Game for Persuasion

We now present a dialogue game for persuasion. As noted above, this game is an instance of the framework of [5], which adapts and further develops the system of [6]. We are particularly interested in using this framework's idea of reply structure on the communication language and its notions of dialogical status of relevance. A crucial feature of our game is that its protocol is flexible in that it allows for alternative replies to moves and for postponement of replies, sometimes even indefinitely. This is important since when an agent sees that a line of attack or defence fails, s/he should be allowed to play other available lines of attack or defence. However, in order to still ensure a strong focus of dialogues this flexibility is constrained by the notion of relevance, to be defined below.

The dialogue game will be presented here in detail since its format plays a crucial role in Section 3 in the combination of the persuasion and negotiation dialogue game. Dialogues are between a *proponent* P and *opponent* O of a single *dialogue topic* $t \in L_t^p$. The game is based on the following ideas. Each dialogue move except the initial one replies to one earlier move in the dialogue of the other party (its *target*). Thus a dialogue can be regarded in two ways: as a sequence (reflecting the order in which the moves are made) and as a tree (reflecting the reply relations between the moves). Each replying move is either an *attacker* or a *surrender*. For instance, a *claim*(p) move can be attacked with a *why*(p) move and surrendered with a *concede*(p) move. And a *why*(p) move can be attacked with an *argue*(A) move where A is an argument with conclusion p, and surrendered with a *retract*(p) move. When s is a surrendering and s' is

Table 2. Speech acts and replies in L_c^p

Acts	Attacks	Surrenders
$claim(\varphi)$	$why(\varphi)$	$concede(\varphi)$
$why(\varphi)$	$argue(A)$ $(conc(A) = \varphi)$	$retract(\varphi)$
$argue(A)$	$why(\varphi)$ $(\varphi \in prem(A))$ $argue(B)$ (B defeats A)	$concede(\varphi)$ $(\varphi \in prem(A)$ or $\varphi = conc(A))$
$concede(\varphi)$		
$retract(\varphi)$		

an attacking reply to s'', we say that s' is an *attacking counterpart* of s. The **persuasion communication language** L_c^p is specified in Table 2. In this table, φ is from L_t^p and arguments A and B are well-formed arguments from $\mathcal{L}$, while defeat relations between arguments are determined according to $\mathcal{L}$. Thus the proof theory of $\mathcal{L}$ is embedded in the persuasion protocol.

The protocol for L_c^p is defined in terms of the notion of a **dialogue**, which in turn is defined with the notion of a **move**.

Definition 1

- *The set M of moves is defined as $\mathbb{N} \times \{P, O\} \times L_c^p \times \mathbb{N}$, where the four elements of a move m are denoted by, respectively:*
  - *$id(m)$, the identifier of the move,*
  - *$pl(m)$, the player of the move,*
  - *$s(m)$, the speech act performed in the move,*
  - *$t(m)$, the target of the move.*
- *The set of dialogues, denoted by $M^{\leq\infty}$, is the set of all sequences $m_1, \ldots,$ $m_i, \ldots$ from M such that*
  - *each i^{th} element in the sequence has identifier i,*
  - *$t(m_1) = 0$;*
  - *for all $i > 1$ it holds that $t(m_i) = j$ for some m_j preceding m_i in the sequence.*

  The set of finite dialogues, denoted by $M^{<\infty}$, is the set of all finite sequences that satisfy these conditions. For any dialogue $d = m_1, \ldots, m_n, \ldots$, the sequence $m_1, \ldots, m_i$ is denoted by d_i, where d_0 denotes the empty dialogue.

When $t(m) = id(m')$ we say that m replies to m' in d and that m' is the target of m in d. We sometimes slightly abuse notation and let $t(m)$ denote a move instead of just its identifier. When $s(m)$ is an attacking (surrendering) reply to $s(m')$ we also say that m is an attacking (surrendering) reply to m'.

The semantics for L_c^p is defined in axiomatic style as a set of precondition-postcondition rules. In fact, as we will see below, the only precondition for each move is that it is legal at this point in the dialogue according to the **protocol**.

Definition 2. *(Protocols for games.) A protocol on M is a set $P \subseteq M^{<\infty}$ satisfying the condition that whenever d is in P, so are all initial sequences that d starts with.*

A partial function $Pr : M^{<\infty} \longrightarrow \mathcal{P}(M)$ is derived from P as follows:

- *$Pr(d) =$ undefined whenever $d \notin P$;*
- *$Pr(d) = \{m \mid d, m \in P\}$ otherwise.*

The elements of $dom(Pr)$ (the domain of Pr) are called the legal finite dialogues. The elements of $Pr(d)$ are called the moves allowed after d. If d is a legal dialogue and $Pr(d) = \emptyset$, then d is said to be a terminated dialogue.

Definition 3. *(The protocol Pr^p for L_c^p.) For all moves m it holds that $m \in P^p(d)$ if and only if m satisfies all of the following rules:*

- *R_1: $pl(m) = T(d)$;*[1]
- *R_2: If $d \neq d_0$ and $m \neq m_1$, then $s(m)$ is a reply to $s(t(m))$ according to L_c^p;*
- *R_3: If m replies to m', then $pl(m) \neq pl(m')$;*
- *R_4: If there is an m' in d such that $t(m) = t(m')$ then $s(m) \neq s(m')$;*
- *R_5: If $d = d_0$, then $s(m)$ is of the form $claim(\varphi)$;*
- *R_6: If $s(m) = retract(\varphi)$, then $C_s(d, m) \not\vdash \varphi$;*
- *R_7: $C_s(d, m)$ is consistent;*
- *R_8: if m is a replying move, then m is relevant in d.*

(for relevance see further below). Further rules could be added, for instance, to prevent circular dialogues [7, 15], but to focus on the essence we will leave such rules undiscussed here.

R_1 says that the player of a move must be to move. R_2-R_4 formalise the idea of a dialogue as a move-reply structure that allows for alternative replies. R_5 says that each dialogue begins with a claim; the initial claim is the *topic* of the dialogue. R_6 requires retractions to be successful and R_7 requires the players to keep their commitments consistent. Finally, rule R_8 says that each replying move must be relevant. This crucial element of the protocol requires some explanation.

Relevance is defined in terms of the *dialogical status* of a move, which in turn is recursively defined in terms of the nature of its replies. A move is *in* iff it is surrendered or else if all its attacking replies are out. (This implies that a move without replies is in). And a move is *out* if it has an attacking reply that is in. With this concept of dialogical status a notion of relevance can be defined. A move is *relevant* if it replies to a relevant target. And a move is a *relevant target* if making it out changes the dialogical status of the initial move of the dialogue. Together with Definition 3 these definitions imply that a move is a relevant target for proponent (opponent) if making it out makes the initial move in (out). Accordingly we say that P *currently wins* d if m_1 is in and O *currently wins* if m_1 is out.

Figure 1 (with only attacking replies) illustrates the notion of relevance. A move labelled + is in and a move labelled − is out. P_5 is not a relevant target

[1] $T(d)$ denotes the player whose turn it is to move in d.

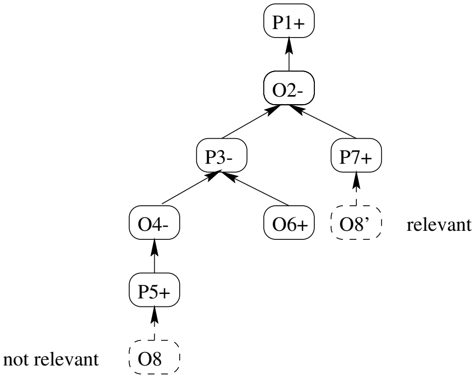


Fig. 1. Relevance of moves

for O: although making P_5 out makes O_4 in, P_3 was already out because of O_6 and therefore O_2 stays out because of P_7, so that P_1 stays in. However, P_7 is a relevant target for O: making P_7 out makes O_2 in since its only attacking reply is now out; then P_1 is out since it now has an attacking reply that is in.

The requirement of relevance comes with a **turntaking rule** T that the turn switches as soon as a player has changed the dialogical status of the initial move (below p is a variable ranging over $\{P, O\}$ and $\overline{p}$ denotes O if $p = P$ and P if $p = O$). Formally, T is a function

- $T : M^{<\infty} \longrightarrow \{P, O\}$

such that $T(d_0) = P$ and if $d \neq d_0$ then $T(d) = p$ iff $\overline{p}$ currently wins d.

The rationale of this rule is that as soon as a player has changed the dialogical status of m_1, he has no relevant moves any more so to avoid premature termination the turn should shift to the other party.

The **effect rules** are defined as a function of the following type:

- C: $\{P, O\} \times M^{<\infty} \longrightarrow \mathcal{P}(L_t^p)$.

$C_p(d)$ denotes the commitments of player p in the dialogue d. The following commitment rules for L_c^p seem uncontroversial and can be found throughout the literature. (Below s denotes the speaker of the move; effects on the other parties' commitments are only specified when a change is effected; finally, d, m stands for the dialogue starting with dialogue d and continuing with move m.)

- If $s(m) = \mathit{claim}(\varphi)$ then $C_s(d, m) = C_s(d) \cup \{\varphi\}$
- If $s(m) = \mathit{why}(\varphi)$ then $C_s(d, m) = C_s(d)$
- If $s(m) = \mathit{concede}(\varphi)$ then $C_s(d, m) = C_s(d) \cup \{\varphi\}$
- If $s(m) = \mathit{retract}(\varphi)$ then $C_s(d, m) = C_s(d)\ - \{\varphi\}$
- If $s(m) = \mathit{argue}(A)$ then $C_s(d, m) = C_s(d) \cup \mathit{prem}(A) \cup \{\mathit{conc}(A)\}$

The **axiomatic semantics** of the system then is as follows: for each move m and dialogue d:

precondition: $m \in Pr^p(d)$
postcondition: as specified by $C_p(d, m)$.

To give a feel for how dialogues evolve in this system, we now list a few properties of the system (see [5] for more details). Firstly, a *turn* of a player always consists of zero or more surrenders followed by a single attack. Further, the turn shifts to the opponent if the initial move is made in while it shifts to the proponent if the initial move is made out. It also follows that a dialogue terminates only if the status of the initial move is *against* the player to move (out for the proponent and in for the opponent). So if a dialogue terminates when player p is to move, p can be said to have *lost* the dialogue. Moreover, it can be shown that a dialogue terminates if and only if either proponent has surrendered to opponent's first move by retracting the dialogue topic or opponent has surrendered to proponent's first move by conceding the dialogue topic. Finally, a 'fairness' and 'soundness' result can be proven about the relation between the dialogical status of the initial move on the one hand and the underlying logic on the other. Under certain conditions the initial move is in just in case the initial claim is defeasibly implied by the 'defended' arguments exchanged by the parties, that is, by the arguments without challenged premises.

Dialogues are not guaranteed to terminate, since the opponent can always continue challenging the proponent's premises. This is a consequence of the fact that the protocol ignores the agents' internal design so that their knowledge bases are not guaranteed to remain fixed during a dialogue. In our opinion this is not a bug but a feature of the protocol since in many realistic settings the agents' knowledge changes during a dialogue. For instance, they may ask advice of third parties, consult databases or make new observations.

3 Negotiation and Argumentation

In the previous sections we introduced protocols for negotiation and persuasion. We now combine them in a way that allows persuasion dialogues to be embedded in negotiation dialogues. In a negotiation dialogue it is the reject move that shows that there is a conflict between the preferences of an agent and the offer that it receives. By starting a persuasion dialogue, the offerer can question the reasons that the offeree has for rejecting. Statements made during persuasion invoke commitments that reflect the preferences of the agents. These commitments are used to restrict further negotiations.

In formally realising the combination of the two dialogue systems, the key idea is to reformulate the negotiation system in the format of Section 2.3 so that the mechanisms of relevance and dialogical status can also be applied to the negotiation part of a dialogue. These mechanisms will then be used to ensure that as long as a persuasion move is legal, no negotiation move can be made: thus the protocol will capture the idea of embedding persuasion in negotiation.

3.1 The Combination

First the combined communication language L_c is defined in Table 3. As can be seen, the negotiation language is reformulated in the format of Section 2.3

Table 3. Speech acts and replies in L_c

Acts	Attacks	Surrenders
negotiation		
$request(\varphi)$	$offer(\varphi')$ $withdraw$	
$offer(\varphi)$	$offer(\varphi'))$ $(\varphi \neq \varphi')$ $reject(\varphi)$ $withdraw$	$accept(\varphi)$
$reject(\varphi)$	$offer(\varphi')$ $(\varphi \neq \varphi')$ $why\text{-}reject(\varphi)$ $withdraw$	
$accept(\varphi)$		
$why\text{-}reject(\varphi_1 \wedge \ldots \wedge \varphi_n)$	$claim(\neg(\varphi_i \wedge \ldots \wedge \varphi_j))$ $(1 \leq i \leq j \leq n)$ $withdraw$	
$withdraw$		
persuasion		
$claim(\varphi)$	$why(\varphi)$	$concede(\varphi)$
$why(\varphi)$	$argue(A)$ $(conc(A) = \varphi)$	$retract(\varphi)$
$argue(A)$	$why(\varphi)$ $(\varphi \in prem(A))$ $argue(B)$ (B defeats A)	$concede(\varphi)$ $(\varphi \in prem(A)$ or $\varphi = conc(A))$
$concede(\varphi)$		
$retract(\varphi)$		

by dividing the “Replies” of Table 1 into surrendering replies ($accept(\varphi)$) and attacking replies (all other replies). Next a new attacking reply is added, viz. $why\text{-}reject(\varphi)$ as a reply to $reject(\varphi)$. The only possible reply to this new locution other than a withdrawal is with a *claim* move from ${L_t}^p$ of which the content negates the conjunction of one or more elements of the rejected offer. Thus the player who rejected the offer can indicate which elements of the offer made him reject it. The use of this reply induces a shift from a negotiation to a persuasion subdialogue.

Next, in order to specify the combined protocol, the notion of negotiation moves must be adapted to fit the format of Definition 1 (which we leave implicit). The combined protocol is then defined as follows.

Definition 4. *(The protocol Pr for L_c.) For all dialogues d and moves m it holds that $m \in Pr(d)$ if and only if m satisfies all of the following rules.*

- R_1*: m satisfies* $R_1 - R_8$ *of Definition 3 but where in* R_2, L_c^p *is replaced by* L_c *and in* R_5, $claim(\varphi)$ *is replaced by* $request(\varphi)$*;*
- R_2*: If* $s(m) = offer(\varphi)$ *and* $s(m_1) = request(\varphi')$ *then* $\{\varphi, \varphi'\}$ *is consistent and* φ *contains at least the same issues as* φ'*;*
- R_3*: If* $s(m) = offer(\varphi)$ *then of no* $m' \in d$, $s(m') = offer(\varphi)$*;*
- R_4*: If* $s(m) = accept(\varphi)$ *then* φ *contains no variables;*

- R_5: *If* m *is a negotiation locution then* m *replies to the most recent target to which a reply is legal;*
- R_6: *If* m *is a negotiation locution then there is no move* $m' \in Pr(d)$ *such that* $s(m')$ *is a persuasion locution;*
- R_7: *If* $s(m) = \mathit{offer}(\varphi)$ *then* $C_s(d) \cup \{\varphi\}$ *and* $C_{\overline{s}}(d) \cup \{\varphi\}$ *are consistent.*

Rule R_1 generalises the general structure of the persuasion protocol to the combined protocol and says that each combined dialogue starts with a request for an offer. Rules $R_2 - R_4$ formalise the negotiation protocol rules of [4] that are not implied by R_1 (see also below). Rule R_5 prevents unnecessary negotiation backtracking moves. Finally, rules R_6 and R_7 perform a key role in the embedding of persuasion in negotiation. R_6 enforces that the relation between the negotiation and persuasion parts of dialogues is one of embedding of the latter in the former (cf. [16]): as long as a persuasion move is legal, no negotiation move is legal. And R_7 formalises the intuition that offers need to respect the reasons for rejection given by the other party when these reasons have been successfully defended in an embedded persuasion dialogue.

Rule R_7 is justified by the following property of the persuasion protocol of [5]: under some plausible assumptions on the contents of arguments a *retract*(t) move in reply to a challenge of the initial claim is always legal. Then by R_6 of the persuasion protocol, which requires retractions to be successful, a player who has defended a rejection with a *claim*(t) move in a terminated persuasion dialogue is committed to t only if he has won the persuasion dialogue about t.

The turntaking rule of the combined system is the same as for persuasion. Given L_c, this rule implies that just as in Section 2.1 the turn shifts after each negotiation move except after an *accept* move, which terminates a dialogue.

Finally, the new commitment rules need to be defined. In fact, they are the same as for persuasion moves in Section 2.3. The effects that negotiation moves have on the players' commitments are irrelevant as long as a dialogue has not terminated, since an offer commits the offeree to an action only after the offer has been accepted: so checking compliance with negotiation commitments lies outside the negotiation dialogue in which the commitment was incurred.

Note that the new system completely preserves the original persuasion system and as much as possible preserves the original negotiation system. Above we already noted that turntaking in the negotiation part is still the same. Furthermore, backtracking from negotiation moves (which was impossible in the original system) is legal in two cases only: if the one who challenges a rejection loses the resulting persuasion dialogue, s/he must move an alternative reply to the rejection, and if the other party loses such a persuasion dialogue, s/he must move a counteroffer or withdrawal in reply to the rejected offer.

3.2 Properties of the Combined Protocol

The main property of the new protocol is about the maximum number of negotiation moves needed to reach a certain agreement.

Proposition 2. *For any proposal φ the maximum length of a negotiation dialogue to end with acceptance of φ is never higher and sometimes lower in the system of Section 3 than in the system of Section 2.1.*

Proof: This follows from the fact that the only effect of a terminated persuasion dialogue on an embedding negotiation dialogue is that it may make offers illegal since they do not respect the commitments of the other agent. Thus the number of legal offers in a negotiation according to Section 3 is never higher and sometimes lower than in a negotiation according to Section 2.1.

Since our persuasion protocol is not guaranteed to terminate, the same holds for our combined protocol. However, on the assumption that a persuasion dialogue always terminates, Proposition 2 implies that the 'success' result on the negotiation protocol proven by [4] still holds for our combined protocol: if the set of possible outcomes is finite then any negotiation is guaranteed to terminate with a withdraw or an accept.

4 An Example

We next illustrate our new protocol by extending our example from Section 2.1 with an embedded persuasion dialogue. For simplicity we paraphrase the contents of the arguments and we do not formally distinguish beliefs, desires and intentions, as is done in e.g. [17, 18]. To illustrate the use of defeasible inference rules, some arguments are assumed to be constructed with presumptive argumentation schemes from [19]. In [20] it is discussed how such schemes can be formalised as defeasible inference rules and their critical questions as pointers to undercutters. Elementary inferences within arguments are paraphrased as *conclusion since premises*. All moves in the dialogue except proponent's last four moves reply to their immediate predecessor.

P_1: $\mathit{request}(\mathit{brand} = ? \wedge \mathit{price} = ?)$
O_2: $\mathit{offer}(\mathit{brand} = \mathit{peugeot} \wedge \mathit{price} = 10000)$
P_3: $\mathit{reject}\ (\mathit{brand} = \mathit{peugeot} \wedge \mathit{price} = 10000)$
Olga now exploits the additional features of the protocol by asking Paul why he rejected the offer.
O_4: $\mathit{why\text{-}reject}(\mathit{brand} = \mathit{peugeot} \wedge \mathit{price} = 10000)$
Paul now meets Olga's challenge of his rejection so that the negotiation shifts into a persuasion. All persuasion moves below until P_{14} reply to their immediate predecessor.
P_5: $\mathit{claim}(\neg\ \mathit{brand} = \mathit{peugeot})$
Paul says that he rejected the offer since he does not want a Peugeot. He is now committed to the content of his claim.
O_6: $\mathit{why}\ (\neg\ \mathit{brand} = \mathit{peugeot})$
P_7: $\mathit{argue}\ (\neg\ \mathit{brand} = \mathit{peugeot}\ \mathit{since}\ \mathit{brand} = \mathit{peugeot} \rightarrow \mathit{brand} = \mathit{french}\ \mathit{and}\ \neg\ \mathit{brand} = \mathit{french})$
It turns out that Paul rejected the offer since a Peugeot is a French car and

he does not want French cars. If Olga now simply concedes Paul's claim as an alternative reply to P_7, the persuasion dialogue terminates and the negotiation is resumed. Then Olga cannot reply to P_3 in the same way as in section 2.1 by offering another french car. Olga could offer a non-French car (as in O_6 in section 2.1) but she chooses to try to persuade Paul that he is wrong in not wanting a French car and she therefore challenges Paul's second premise.

O_8: *why* ($\neg$ *brand* = *french*)

P_9: *argue* ($\neg$ *brand* = *french since having french cars is bad; this is so since french cares are unsafe and having an unsafe care usually is bad.*)

Paul defends his second premise with an argument from (bad) consequences.

O_{10}: *why* (*french cars are unsafe*)

P_{11}: *argue* (*french cars are unsafe since car magazine mycar says so and mycar are experts about cars*)

This is a defeasible argument based on the argumentation scheme from expert opinion: "what experts say is normally true".

O_{12}: *argue* (*magazine mycar is biased since magazine mycar is german and german car magazines are usually biased against french cars*)

Using a default rule, Olga constructs an undercutter of the argument from expert opinion, namely that this expert is biased.

P_{13}: *concede* (*magazine mycar is german*)

P_{14}: *concede* (*german car magazines are usually biased against french cars*)

Even though Paul has conceded the premises of Olga's undercutter, he can still move a counterargument, since the argument is defeasible because it uses a default rule. Paul chooses to rebut the undercutter, using another default rule.

P_{15}: *argue* ($\neg$ *magazine mycar is biased since magazine mycar has a very high reputation and car magazines with high reputation usually are not biased*)

Note that P_{14} is a second and P_{15} a third reply to O_{12}.

O_{16}: *why* (*magazine mycar has a very high reputation*)

Let us assume that Paul now realises that he has no plausible way to defend his premise that the car magazine has a high reputation. At this point, all of Olga's persuasion moves are relevant targets for Paul. He could, for instance, move another rebuttal of Olga's undercutter, or another argument why French cars are not safe or why he does not want french cars. But suppose that Paul sees no plausible way of doing so and instead retracts that he does not want French cars by moving an alternative reply to O_8 and then retracting his main claim as a second reply to O_6.

P_{17}: *retract* ($\neg$ *brand* = *french*)

P_{18}: *retract* ($\neg$ *brand* = *peugeot*)

Now Paul has no legal persuasion moves any more since all targets have become irrelevant: since Paul has surrendered to O_6, his main claim P_5 cannot be changed from *out* to *in*. So the persuasion dialogue terminates and the negotiation resumes with Olga to move after P_2. Since with P_{18} Paul has ended his commitment to his main claim, Olga is now allowed to offer another French car, perhaps even a Peugeot for a lower price. The negotiation could now continue as in Section 2.1 with move O_4.

It is instructive to construct the dialectical graph of arguments and counterarguments exchanged by Paul and Olga during the persuasion dialogue ($p \rightsquigarrow q$ reads as "if p then usually q").

The graph contains a simple argument game according to the proof theory of the underlying logic. Since on the basis of the information exchanged during the persuasion dialogue no other counterarguments to one of these three arguments can be constructed, the graph is actually a proof that, on the basis of this information, the proposition $\neg$ *peugeot* is justified. However, the last argument in the graph has one challenged premise, viz. *highrep*, so this argument is not defended (indicated by the dotted box). The defended part of the graph is instead a proof that $\neg$ *peugeot* is not justified on the basis of all *defended* information.

5 Conclusion

In this paper we have presented a protocol for negotiation with embedded argumentation that has a social semantics. In doing so, we have exploited the general format of [6, 5] of dialogue systems. In the resulting dialogue game reasons for rejections can be asked and, when given, can constrain further offers unless the offering agent can persuade the rejecting agent that the reason is not tenable. Thus a negotiation is sometimes more efficient since offers that will certainly be rejected can be avoided, and it is sometimes of higher quality since flawed reasons can be revised. The persuasion protocol is flexible in that it allows for different underlying logics, for alternative replies and for postponing replies, sometimes even indefinitely. Yet a strong focus of dialogues is maintained through the requirement of relevance.

We know of one earlier protocol that allows for persuasion dialogues about rejections, viz. [3]. It was a source of inspiration for the present work but there are reasons for further development. The first is that the protocol does not have

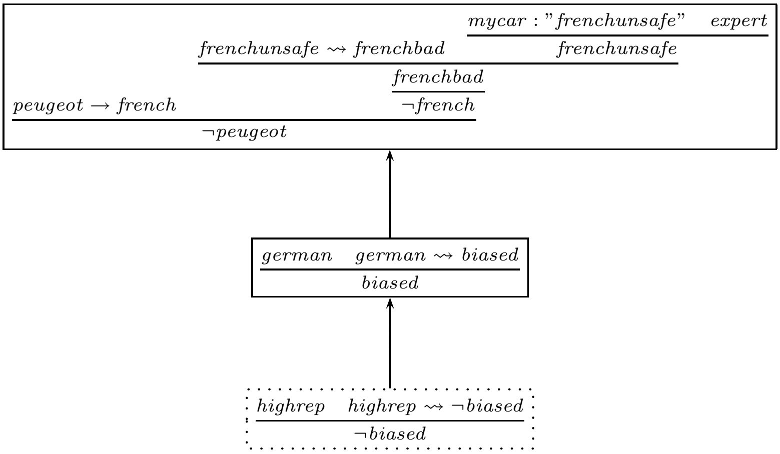


Fig. 2. The dialectical graph

a social semantics, since whether an agent is allowed to assert a claim or an argument partly depends on their internal mental state. Also, arguments have to be classical propositional proofs from a consistent set of premises so that, for instance, the use of presumptive argument schemes or undercutting counterarguments is not supported. A further limitation is that the dialectical aspects of the underlying logic are only used internally by an agent, to verify whether they have an acceptable argument for a claim in their (possibly inconsistent) knowledge base. By contrast, in our protocol the dialectical role of each argument in a dialogue is made explicit, as illustrated by Figure 2. Finally, the protocol of [3] only weakly maintains focus of dialogues, allowing, for example, dialogues like P_1: $claim(p)$, O_2: $why(q)$.

In future research the present protocol should be combined with relevant other work. For instance, [17, 18] define a rich topic language in which the beliefs, desires and intentions of agents can be distinguished and reasoned about, allowing negotiating agents to produce and attack several interesting types of arguments. Since we have partially abstracted from the nature of the persuasion topic language and defeasible inference rules, our dialogue system can be instantiated with this work. We also aim to study the interaction of the present protocol with agent designs and strategies, as, for instance, done in [8] for several dialogue types. Finally, we aim to include other forms of argument-based negotiation, such as arguments why a proposal should be accepted ([17, 18]).

Acknowledgements

Henry Prakken was partially supported by the EU under IST-FP6-002307 (ASPIC). The authors thank Birna van Riemsdijk for her useful comments on a draft of this paper.

References

1. Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review **18** (2003) 343–375
2. Singh, M.: Agent communication languages: rethinking the principles. IEEE Computer **31** (1998) 40–47
3. Amgoud, L., Parsons, S., Maudet, N.: Arguments, dialogue, and negotiation. In: Proceedings of the Fourteenth European Conference on Artificial Intelligence. (2000) 338–342
4. Wooldridge, M., Parsons, S.: Languages for negotiation. In: Proceedings of the Fourteenth European Conference on Artificial Intelligence. (2000) 393–400
5. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. Journal of Logic and Computation **15** (2005) to appear.
6. Prakken, H.: On dialogue systems with speech acts, arguments, and counterarguments. In: Proceedings of the 7th European Workshop on Logic for Artificial Intelligence (JELIA'2000). Number 1919 in Springer Lecture Notes in AI, Berlin, Springer Verlag (2000) 224–238

7. Walton, D., Krabbe, E.: Commitment in Dialogue. Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY (1995)
8. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. Journal of Logic and Computation **13** (2003) 347-376.
9. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In Gabbay, D., Günthner, F., eds.: Handbook of Philosophical Logic. Volume 4. Second edn. Kluwer Academic Publishers, Dordrecht/Boston/London (2002) 219–318
10. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n–person games. Artificial Intelligence **77** (1995) 321–357
11. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-classical Logics **7** (1997) 25–75
12. Amgoud, L., Cayrol, C.: A model of reasoning based on the production of acceptable argument. Annals of Mathematics and Artificial Intelligence **34** (2002) 197–216
13. Pollock, J.: Cognitive Carpentry. A Blueprint for How to Build a Person. MIT Press, Cambridge, MA (1995)
14. Dung, P.: Logic programming as dialog game. Unpublished paper, Division of Computer Science, Asian Institute of Technology, Bangkok (1994)
15. Mackenzie, J.: Question-begging in non-cumulative systems. Journal of Philosophical Logic **8** (1979) 117–133
16. McBurney, P., Parsons, S.: Games that agents play: A formal framework for dialogues between autonomous agents. Journal of Logic, Language and Information **13** (2002) 315–343
17. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence **104** (1998) 1–69
18. Parsons, S., Sierra, C., Jennings, N.: Agents that reason and negotiate by arguing. Journal of Logic and Computation **8** (1998) 261–292
19. Walton, D.: Argumentation Schemes for Presumptive Reasoning. Lawrence Erlbaum Associates, Mahwah, NJ (1996)
20. Bex, F., Prakken, H., Reed, C., Walton, D.: Towards a formal account of reasoning about evidence: argumentation schemes and generalisations. Artificial Intelligence and Law **12** (2003) 125–165

New Types of Inter-agent Dialogues*

Eva Cogan[1], Simon Parsons[1], and Peter McBurney[2]

[1] Deptartment of Computer and Information Science
Brooklyn College
City University of New York
2900 Bedford Avenue
Brooklyn, NY 11210 USA
{cogan, parsons}@sci.brooklyn.cuny.edu
[2] Department of Computer Science
University of Liverpool
Chadwick Building
Peach Street, Liverpool L69 7ZF, UK
p.j.mcburney@csc.liv.ac.uk

Abstract. Much work in the area of argumentation-based dialogues between agents has been based on the influential taxonomy of dialogue types developed by Walton and Krabbe. In this paper we re-examine the Walton and Krabbe framework, concentrating on the preconditions for different types of dialogue and analyzing them in a systematic way. Doing so identifies a number of new kinds of dialogue missing from the framework. We discuss some of the more interesting of these and develop protocols for them.

1 Introduction

Sycara [26, 27] has reasonable claim to be the first to suggest the use of *argumentation* in inter-agent dialogues, that is: the exchange of reasons in favor of and against the assertions of dialogue participants. There has been increasing interest in, and work on, the use of argumentation-based techniques . An important influence on this work was a paper by Reed [24] which introduced the work of the philosophers Walton and Krabbe [28] to researchers interested in this form of dialogue. Walton and Krabbe distinguish six basic forms of dialogue :

Information seeking: one participant seeks the answer to some question(s) from another participant, who is believed by the first to know the answer(s).

Inquiry: participants collaborate to answer some question(s) whose answer(s) are not known to any one participant.

Persuasion: one participant seeks to persuade the other participant to adopt a belief or point-of-view that the second does not currently hold.

Negotiation: participants bargain over the division of some scarce resource.

* This paper is a modified version of E. Cogan, S. Parsons, and P. McBurney. What kind of argument are we going to have today? In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *Proceedings of the Fourth International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 544-551. ACM Press, 2005.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 154–168, 2006.

Deliberation: participants collaborate to decide what course of action to take.
Eristic: participants quarrel verbally as a substitute for physical fighting.

A number of authors have taken Walton and Krabbe's framework as a starting point for discussing various kinds of inter-agent dialogue. For example, [4, 12, 23] have discussed *persuasion*, [17] considered *inquiry*, [16, 19] looked at *negotiation*, and [15] examined *information seeking*. [20, 21], defined simple protocols for, and investigated the properties of, persuasion, information seeking, and inquiry dialogues. Others, [6, 11, 25, 29] for example, investigated types of dialogue that are not covered by Walton and Krabbe (who make no claims of comprehensiveness).

Our long term aim is to identify and formalize a set of dialogue types that will support a wide range of agent interactions. Along with the dialogue types, we seek protocols that agents can follow to engage in these dialogues. Towards this aim, in this paper we take a systematic approach to analyzing dialogue preconditions, and identify a set of new dialogue types. This moves us some way towards a comprehensive classification that will allow agents to select from a broad range of dialogue types to best suit their dialogical needs.

2 Background

2.1 Argumentation

We start with the formal system of [20, 21], which we present very briefly and informally. A full, formal description is in [20, 21]. That system also deals with preferences between arguments, which, for simplicity, we ignore here.

Each agent involved in a dialogue has a knowledge base which contains formulas of a propositional language $\mathcal{L}$. $\vdash$ stands for classical inference and $\equiv$ for logical equivalence. An *argument* is a pair $A = (S, p)$ where p is a formula of $\mathcal{L}$ and S a consistent subset of the knowledge base such that $S \vdash p$ and no proper subset of S does so. S is called the *support* of A, written $S = \text{Support}(A)$, and p is the *conclusion* of A, written $p = \text{Conclusion}(A)$.

Two arguments may conflict. More precisely, arguments may *undercut* one another, where argument A_1 undercuts A_2 iff $\exists p \in \text{Support}(A_2)$ such that $\neg p \equiv \text{Conclusion}(A_1)$. In other words, an argument is undercut if and only if there is another argument which has as its conclusion the negation of an element of the support for the first argument. There are, of course, other ways to define a system of argumentation. This is just one approach, based on [1, 2], which itself is based on [7], and which our experience suggests is an adequate framework for handling agent communication.

Now, a set of arguments $\mathcal{S}$ *defends* an argument A iff for each argument B that undercuts A, there is an argument in $\mathcal{S}$ that undercuts B. From this notion we can develop the important idea of an *acceptable* argument. An acceptable argument A is one that is not undercut, or for which there is an acceptable argument that undercuts each of the arguments that undercut A. An acceptable argument is one which is, in some sense, proven since all the arguments which might undermine it are themselves undermined. However, this status can be revoked following the discovery of a new argument (possibly as the result of the communication of some new information from another agent).

Table 1. Notation

$CS(X)$	the commitments made by the agent during the current dialogue
$B_X p$	p is the conclusion of an acceptable argument
$I_X p$	$p \in INT(X)$ (Intentions)
$W_X p$	$B_X p \ \vee \ B_X \neg p$ (Whether)
$A_{X,Y} p$	$(B_X p \wedge B_Y p) \vee (B_X \neg p \wedge B_Y \neg p)$ (Agree)

2.2 Agents and Dialogue

We build a model of dialogue on top of this system of argumentation. Dialogues take place between two agents. Each agent X has a private knowledge base. One part of this, the belief base $BEL(X)$, contains the agent's beliefs. In addition, each agent has a set of obligations, intentions and desires, denoted by $OBL(X)$, $INT(X)$ and $DES(X)$ respectively, which are modeled as a multi-context system as in [19]. Such a system can take care of nested modalities and the necessary constraints between modalities, as described in [19]. Our agents are BOID agents in the sense of [5] — though we don't *require* obligations or desires for the work described here, we keep them in the model for continuity with our other work. We write $I_X p$ to denote $p \in INT(X)$ and $B_X p$ to denote that p is the conclusion of an acceptable argument[1]. Each agent's commitment store, $CS(X)$, contains the commitments made by the agent during the current dialogue. Following Hamblin [14] we take commitments to be propositions that an agent is prepared to defend. Each agent in a dialogue has access to its own private knowledge base and both commitment stores. The union of the commitment stores can be viewed as the public state of the dialogue at any given time.

We further define two notions:

- $W_X p$ (X knows whether or not p) which denotes $B_X p \ \vee \ B_X \neg p$.
- $A_{X,Y} p$ (X and Y agree about p)which denotes $(B_X p \wedge B_Y p) \vee (B_X \neg p \wedge B_Y \neg p)$.

Table 1 summarizes the notation used in this paper.

2.3 Locutions

The following locutions (moves in the dialogue game) are available to the agents. Some of the moves we use here were first introduced in [20] and modified in [22]. Each locution has a rule describing how to update the commitment store after the move. For all moves, player G addresses the ith move of the dialogue to player H, p is a proposition, and S is a set of propositions. The special character $\mathcal{U}$ may also be asserted. It indicates that G cannot give an answer. As soon as $\mathcal{U}$ is asserted, the dialogue terminates.

The first two moves allow propositions to be asserted. An agent uses these locutions to state propositions that it wishes to place "on the record" in the dialogue. Typically these are ones that it wishes the other agent in the dialogue to accept. The next two moves respond to assertions, taking the propositions that another agent has asserted and moving them into the speaker's commitment store. The $question$ locution can be used to ask the other player about the truth of any proposition. Since a question makes no

[1] Any proposition p in $\mathcal{L}$ is the conclusion of an argument $(\{p\}, p)$.

commitment, the CS remains unchanged. Finally, $challenge$ is a means of asking the other player to state the support of an argument for a proposition.

$$\begin{array}{ll}
assert(p) & CS_i(G) = CS_{i-1}(G) \cup \{p\} \\
assert(S) & CS_i(G) = CS_{i-1}(G) \cup S \\
accept(p) & CS_i(G) = CS_{i-1}(G) \cup \{p\} \\
accept(S) & CS_i(G) = CS_{i-1}(G) \cup S \\
question(p) & CS_i(G) = CS_{i-1}(G) \\
challenge(p) & CS_i(G) = CS_{i-1}(G)
\end{array}$$

The preconditions for the locutions are determined by what has previously been called the *attitude* of an agent and the content of the agent's knowledge base. While a range of such attitudes are explored in [22], here we restrict ourselves to considering what [22] call a thoughtful/skeptical agent; that is, one who is allowed to $assert$ and $accept$ only propositions for which it has an acceptable (in the sense defined above) argument. Such preconditions do not uniquely define which locutions an agent can use at a particular point in time. Additional constraints are provided by a protocol. Examples of the kind of protocol that we are interested in are given in [20].

2.4 Dialogue Protocols

As mentioned above, [20] introduced some simple dialogue protocols. In order to contrast those with the ones we introduce here, we restate the protocols from [20]. In addition, we formalize the preconditions that [20] states informally.

Before giving these protocols, however, we first define a macro $CD(X, Y, p)$ for a common set of locutions used in the "challenge and defense" of a proposition. Suppose agent X has asserted a proposition p for which agent Y has no acceptable argument. Agent Y then challenges p. Agent X attempts to defend p by providing the support of an argument for p. Y may then (when necessary) challenge each element of the defense. If Y accepts the elements of the defense and they do indeed form the support of an acceptable argument for p, then Y can accept p.

$CD(X, Y, p)$

1. Y *challenge*s p
2. $\begin{cases} X \ assert\text{s } S, \text{the support of an argument for } p & \text{if allowed by its attitude,} \\ \text{the dialogue terminates} & \text{otherwise.} \end{cases}$
3. for each $s \in S$ $\begin{cases} Y \ accept\text{s } s & \text{if allowed by its attitude,} \\ CD(X, Y, s) & \text{otherwise.} \end{cases}$
4. $\begin{cases} Y \ accept\text{s } p & \text{if allowed by its attitude,} \\ \text{the dialogue terminates} & \text{otherwise.} \end{cases}$

We now give the protocols from [20] using $CD(X, Y, p)$. The preconditions are drawn up from the perspective of G, the agent that utters the first locution in any dialogue using the protocols.

$Information\ seek(G, H, p)$

preconditions:

- $\neg W_G p$
- $I_G W_G p$
- $\neg B_G \neg W_H p$

locutions:

1. $G\ questions(p)$
2. $\begin{cases} H\ asserts\ p & \text{if allowed,} \\ H\ asserts\ \neg p & \text{if allowed,} \\ H\ asserts\ \mathcal{U} & \text{otherwise.} \end{cases}$
3. $\begin{cases} G\ accepts\ H\text{'s response} & \text{if allowed,} \\ CD(H, G, H\text{'s response}) & \text{otherwise.} \end{cases}$

$Inquiry(G, H, p)$

preconditions:

- $\neg W_G p$
- $I_G A_{G,H} p$
- $B_G \neg W_H p$
- $B_G I_H A_{G,H} p$

locutions:

1. G asserts $q \rightarrow p$ for some q, or $\mathcal{U}$.
2. $\begin{cases} H\ accepts\ q \rightarrow p & \text{if allowed,} \\ CD(G, H, q \rightarrow p) & \text{otherwise} \end{cases}$
3. $H\ assert$s q, or $r \rightarrow q$ for some r, or $\mathcal{U}$.
4. $\begin{cases} G\ accepts\ H\text{'s assertion} & \text{if allowed,} \\ CD(H, G, H\text{'s assertion}) & \text{otherwise} \end{cases}$
5. If $\mathcal{A}(CS(G) \cup CS(H))$ (the set of arguments that can be made from $CS(G) \cup CS(H)$) includes an argument for p which is acceptable to both agents, then first G and then H accepts p and the dialogue terminates successfully.
6. Go to 3, reversing the roles of G and H and substituting r for q and some t for r.

$Persuade(G, H, p)$

preconditions:

- $B_G p$
- $B_G \neg B_H p$
- $I_G B_H p$

locutions:

1. $G\ asserts\ p$
2. $\begin{cases} H\ accepts\ p & \text{if allowed,} \\ H\ asserts\ \neg p & \text{if allowed,} \\ CD(G,H,p) & \text{otherwise.} \end{cases}$
3. If $H\ asserts\ \neg p$, then go to 2 with the roles reversed and $\neg p$ instead of p.

We now turn to the main contribution of this paper: by examining the preconditions of dialogues, we identify new kinds of dialogue and specify protocols for them.

3 Preconditions for Dialogue

We summarize Walton and Krabbe's [28, pages 65–85] descriptions of the three dialogue types that deal with beliefs (which will be our focus here) as:

Information Seeking Dialogues: One participant has some information, or is in a position to know it, and the other both does not have the information and needs it. Both participants share the goal of spreading knowledge.

Inquiry Dialogues: The participants collaborate to answer some question(s) whose answer(s) are not known to any one participant. Both parties are initially ignorant about the answer(s), but are committed to resolving the question(s).

Persuasion Dialogues: These dialogues begin with one participant supporting a particular statement which the other participant does not hold, and the first seeks to resolve the conflict by convincing the second to adopt the proposition. The second party shares the objective of resolving the conflict, but may try to do this by convincing the other to change his or her mind.

One way to interpret Walton and Krabbe's descriptions is in terms of the conditions that hold at the beginning and end of a specific kind of dialogue. In the literature this has typically been done in the sense of defining the initial conditions that any dialogue protocol must cope with, and the final conditions it must bring about to be successful (for example as in [3]). Thus, since an inquiry dialogue starts with no one participant knowing whether or not the proposition in question is true, and will end successfully with a proof of the proposition or its negation, the minimum requirement for an inquiry dialogue is that it must be able to construct a proof where the various components are distributed among the participants, exactly as in the Inquiry protocol given above.

Another approach, and the one we pursue here, is to consider the preconditions as a guide to the participants as to what kind of dialogue is appropriate. Thus if Shimon doesn't know *whether* p and needs to know, then if Piotr knows *whether* p, it makes sense for Shimon to engage Piotr in an information seeking dialogue, but if Piotr does not know *whether* p, then it makes sense for Shimon to engage him in an inquiry into p. From this perspective, we can think of Table 2 (which is taken from [13] and modified to mesh with our notation) as specifying which dialogue is appropriate under which conditions. Here, as for all the tables in this paper, the preconditions are laid out along

Table 2. Preconditions from Walton and Krabbe [13, 28]

	$B_H p$	$B_H \neg p$	$\neg W_H p$
$B_G p$		Persuasion	InfoSeek
$B_G \neg p$	Persuasion		InfoSeek
$\neg W_G p$	InfoSeek	InfoSeek	Inquiry

Table 3. Modified preconditions from Walton and Krabbe

	$B_G B_H \neg p \;\; B_G I_H A_{G,H} p$	$B_G \neg W_H p \;\; B_G I_H A_{G,H} p$	$B_G W_H p$
$B_G p \;\; I_G A_{G,H} p$	Persuasion		
$\neg W_G p \;\; I_G A_{G,H} p$		Inquiry	InfoSeek

both axes and the cell contains the relevant dialogues. If the dialogue is successful, the intention(s) of the participants will be fulfilled. A space indicates that there is no dialogue that applies. For example, in Table 2 there is no dialogue when both agents agree on the truth of a proposition.

However, though this characterization is neat and apparently faithful to what Walton and Krabbe intended, it is no use to Shimon in his efforts to determine what kind of dialogue is appropriate in determining the truth of p. Why not? Because he will not generally know the truth of $B_{Piotr} p$. He can determine only if $B_{Shimon} B_{Piotr} p$ — and must use this to make his decision about the most appropriate dialogue.

Furthermore, the goal (or, as we model it here, the intention) of the participants comes into play. It is not just their mutual ignorance about p that suggests Shimon should engage Piotr in an inquiry, but the fact that Shimon intends to know whether p is true or not, and believes that Piotr does the same. These considerations suggest that Table 3 rather than Table 2 is what Shimon should use to determine what kind of dialogue is most appropriate. It takes the goals of the dialogue, as stated in [28] and restates them as preconditions.

Note that Table 3 deals only with the conditions from G's perspective (in other words in terms of G's beliefs). Like the remainder of the tables in this paper, it presents the perspective of the initiator of the dialogue. It also exploits the symmetry in p and $\neg p$. Were we to distinguish p and $\neg p$ in G's beliefs we would get an expanded version. We could further add a similar set of results for H and get a table that includes all the entries in Table 2. We leave these additional entries out here and for the remainder of the paper because they are redundant.

The table does more than tell Shimon what dialogues are appropriate in different situations: it identifies some suggestive gaps. For example, under Walton and Krabbe's definition, it isn't possible for G to engage H in a persuasion unless H wants to resolve the inconsistency. If H doesn't care, then the dialogue cannot be a persuasion. We argue that this is unnecessarily restrictive. We have all been party to persuasions where we didn't want to resolve the issue but were forced into the dialogue by some convention (reluctant encounters with authority for example, or not wishing to give too much offence to doorstopping evangelists) and from the point of view of formalization, actively requiring both participants to want to resolve the situation isn't necessary.

Table 4. Modified preconditions from Parsons, Wooldridge and Amgoud [22]

		$B_G B_H p$	$B_G B_H \neg p$	$B_G \neg W_H p$	$B_G \neg W_H p$	$B_G W_H p$
				$B_G I_H A_{G,H} p$		
$B_G p$	$I_G B_H p$		Persuasion	Persuasion	Persuasion	
$\neg W_G p$	$I_G A_{G,H} p$	InfoSeek	InfoSeek	Inquiry		InfoSeek

Provided that H is at least *cooperative*, in the sense of not actively trying to derail or prolong the dialogue[2], then G may rationally initiate a persuasion.

Examining the protocols given above reveals that it is possible to relax the preconditions for persuasion and information seeking. In particular, $I_G A_{G,H} p \wedge B_G I_H A_{G,H} p$ (a requirement in [20]) is not required for G to initiate a persuasion. Under the interpretation we favor, a sufficient condition for G to start a persuasion is that $I_G B_H p$, G wants H to believe p. Indeed, the protocol for persuasion given above also works when $B_G \neg W_H p$, that is whether or not H believes anything about p.

For an information seeking dialogue, we suggest that the preconditions should allow G to start a dialogue whether or not G believes it knows H's position on p, as long as $\neg W_G p$ (G doesn't currently have a position on p). Once again, the protocol works under these conditions, and it seems a sensible relaxation. Some information seeking dialogues make sense under such conditions. For example, when Shimon is hopelessly lost, he might randomly ask people in the street for directions. He doesn't know whether they know the place to which he is headed, but he might still want to ask them.

With these new preconditions, Table 3 expands to become Table 4. Note that the preconditions given are not those as stated in [20], but are consistent with the dialogues given there.

4 New Dialogues and Protocols

Despite this relaxation of the initial conditions, there remain several situations in which it seems natural to engage in dialogues, but to which the basic Walton and Krabbe dialogue types do not apply. In this section we identify some of these situations and give protocols that capture them, extending the set of protocols given in [20]. Note that we are not claiming to identify all possible dialogues here (one could, of course, continue modifying preconditions more or less forever). Rather, by carefully considering the preconditions, we can identify some useful kinds of dialogue that are apparently not included in the Walton and Krabbe classification (dialogues, therefore, that couldn't take place under a strict implementation of the Walton and Krabbe typology).

To start with, we note that, as things stand, an agent is allowed to engage in information seeking and inquiry dialogues only if it is ignorant (to use Walton and Krabbe's [28, page 66] terminology) about the subject of the dialogue. The only kind of dialogue about p in which one can engage when one knows p, according to Walton and Krabbe,

[2] [8, 9, 10] give examples of cases where one would not want to be so coooperative, for example when engaged in a dialogue with law enforcement officers who wish to persuade one to confess to a crime.

is persuasion. However, there are cases in which it is natural to have other kinds of dialogue about some p that one believes to be true.

Consider that Shimon believes some proposition p to be true (p might be the proposition that "According to Walton and Krabbe, both participants in a persuasion dialogue have to start the dialogue with opinions about the subject of the dialogue"), but wants to check whether he is correct by asking Piotr if he thinks this is the case. This would be an information seeking dialogue if Shimon didn't already have an opinion about p. Since the initial conditions differ from an information seeking dialogue, we require a new dialogue type and a new protocol. We call this kind of dialogue a *verification* dialogue.

4.1 Verification Dialogue

In a verification dialogue, agent G seeks the answer to some question from agent H. The proposition with which the dialogue is concerned is p. Unlike information seeking, verification no longer requires that G doesn't know p ($\neg W_G p$). It requires only that G wants to see if H thinks p is true ($I_G B_G B_H p$), and we don't have any condition on what H believes or on what G believes that H believes (we are all familiar with dialogues in which we ask, for instance, "Do you want that last piece of cake?", thinking the answer will be "yes", but hoping it will be "no" and these seem to be verification dialogues just as much as the previous example). One possible protocol for conducting a verification dialogue about p is the following. Note that all the protocols given in this paper, like those in [20], are the minimal protocol we can imagine for the task at hand.

$\mathit{Verify}(G, H, p)$

preconditions:

- $I_G B_G B_H p$

locutions:

1. $G\ questions\ p$
2. $$\begin{cases} H\ asserts\ p & \text{if allowed,} \\ \text{dialogue terminates unsuccessfully} & \text{otherwise.} \end{cases}$$

If H asserts p, the dialogue was successful.

If the dialogue fails and G wants to continue the discussion about p, G must initiate another dialogue. For example, G might then proceed to *persuade* H. Since a verification dialogue is narrowly focused on the question of whether H believes p or not, it is even simpler than an information seeking dialogue (which requires that G be sure to check the grounds of H's argument for p in order to know whether it can accept p). But a verification dialogue won't help G if it wants to know the *reason* that H believes p.

Knowing the reason may be irrelevant — as when Shimon just wants to check his facts about Walton and Krabbe. However, knowing the reason may be important. Shimon may have an argument for "It is important to attend AAMAS this year" based on the fact that his friends will be there, but want to come up with a stronger one if possible (say, to convince the chairman of his department to pay for the trip). As a result Shimon may want to find out Evelyn's reason for the importance of AAMAS, in case it is a better argument. Similarly, Shimon may be about to engage Piotr in a persuasion about p ("Shimon deserves to be the lead author of the paper Shimon and Piotr are writing"), and might think his chances of convincing Piotr are improved if he first learns Piotr's reasons for Piotr's position. (Shimon can then construct an argument that is less likely to be undercut.) In either case, we need a form of dialogue which focuses on the argument rather than the subject itself. We call this kind of dialogue a *query*, and describe it in detail next.

Another common example, as suggested in [25], which distinguishes between *verify* and *query* is that of a teacher who asks a student a question to which the teacher already knows the answer. The teacher is looking to *verify* that the student knows the answer as well. If the teacher wants the student to defend his position, it becomes a *query*.

4.2 Query Dialogues

The query dialogue arises in a situation where G will $always\ challenge$ after $H\ assert$s its answer about p because G isn't interested only in whether or not H believes p, but rather in H's argument for p. This marks a shift from the underlying assumptions used in introducing the protocols in [20], where agents always $accept$ed whenever their attitude allowed. For a query, agents always $challenge$. A simple protocol for a query dialogue is as follows:

$Query(G, H, p)$

preconditions:

- $I_G W_G p$
- $\neg B_G \neg W_H p$

locutions:

1. $G\ questions\ p$
2. $\begin{cases} H\ asserts\ p & \text{if allowed,} \\ \text{dialogue terminates unsuccessfully} & \text{otherwise.} \end{cases}$
3. $CD(H, G, p)$

A dialogue under the Query protocol[3] succeeds when H offers an argument for p that is acceptable to G. Note again that we don't require G to be ignorant about p before undertaking the dialogue.

We consider that the dialogue has failed if G doesn't find H's argument acceptable since it has failed in its objective of discovering an argument. G's perspective is the only one that counts here because G initiated the dialogue. However, this does not mean that the dialogue need have been a waste of time for G. At the very least G may have obtained some new information (some of H's grounds) that G can use to construct a

[3] We will follow the convention of referring to a dialogue under a specific protocol by the name of the protocol, so that a Query dialogue is one under the Query protocol, and is distinct from a "query dialogue", which is any dialogue in the general class in which one agent is interested in the argument another has for a proposition.

different, new argument. Furthermore, if G started the query to discover H's argument prior to a persuasion, then a failure might be more helpful to G than a success.

This completes our discussion of Query dialogues, but there is another kind of dialogue that stands in the same relation to those generated by the Query protocol as inquiry does to information seeking. Under the conditions proposed by Walton and Krabbe, an inquiry can take place only when both agents don't know whether or not p is true, and both intend to resolve the matter. There is another kind of query, a mutual query, in which G and H work together to establish a mutually acceptable argument for p, but from a position that either or both of them already have an opinion about the truth of p. Such a dialogue has some elements of persuasion and inquiry as defined by Walton and Krabbe, but we believe it to be subtly different enough to be a separate class of dialogue.

An example here is when Shimon and Evelyn get together to discuss their ideas for a paper on new kinds of dialogue. Evelyn believes that they have a new classification of dialogue types and wants to check that Shimon agrees. Now, because this isn't something that Shimon has necessarily thought about prior to the meeting, Evelyn can't just $question$ and launch into a query dialogue. In addition, Evelyn can't use an inquiry, since that requires her to not believe she has a classification before the dialogue commences. Furthermore, it isn't a persuasion because what is important is not Evelyn convincing Shimon to agree, but seeing whether they can *jointly* build a case. Instead, what is required is a dialogue in which the two of them jointly construct the case for writing the paper, arguing out the truth of each step along the way, while allowing Evelyn to have a position on the subject of the dialogue before the dialogue commences.

To cover this case we introduce a $Query\mathbf{2}$ dialogue which does exactly this. One possible protocol for it is:

$Query\mathbf{2}(G, H, p)$

preconditions:

- $I_G W_G p$
- $B_G \neg W_H p$

locutions:
The protocol then proceeds as for Inquiry (Sect. 2.4).

This completes our discussion of $Query\mathbf{2}$, but there is yet another kind of query dialogue that we can imagine.

Going back to the case of Shimon and Evelyn's discussion about writing a paper which motivated the Query**2** protocol, we recall that it started from the position that Evelyn wanted to discuss *whether* they had a new classification of dialogue types. We can easily imagine a situation in which Evelyn hopes that Shimon and Evelyn together might produce an acceptable argument*for* p (in other words an argument that proves p is true), rather than aiming to know the truth of p.

Thus Evelyn may initiate this dialogue irrespective of either participant's current position on p. In fact, it even makes sense to initiate this kind of dialogue when either or both participants believe the proposition to be false. Although Evelyn and/or Shimon might initially believe that there isn't a paper to be written, the discussion might end up constructing an argument for the proposition that there is one.

This seems to us to be a new kind of query dialogue, one we will call $Query\mathbf{3}$, and a protocol for such a dialogue is:

$Query\mathbf{3}(G, H, p)$

preconditions:

- $I_G B_G p$

locutions:
The protocol then proceeds as for Inquiry (Sect. 2.4).

An interesting kind of dialogue that is close to Query**3** is one in which a criminal lawyer and a defendant jointly seek arguments to prove that the defendant is innocent, whether or not they individually believe this to be the case. The lawyer's job in such a case is not to determine whether or not his client committed the crime but to produce a good case for the defense. He wants the dialogue to produce an argument to convince the jury that the defendant didn't commit the crime.

This completes our discussion of new dialogue types.

4.3 A New Classification

With these new kinds of dialogue, we can fill in the gaps in Table 4. In fact, we do more than that: we cover interactions which (in terms of their preconditions) were obscured in the previous tables, identifying new goals that G might have for engaging in a dialogue. The result is Table 5.

Furthermore, Table 5 reflects some subtle changes to inquiry and persuasion dialogues as well. In persuasion dialogues, we weaken the condition on G's beliefs about H's beliefs about p so that G can engage H in a persuasion without even knowing that H doesn't agree about p (on top of the previous relaxation that G no longer had to know that H disagrees). Now the key thing is that G believes p and wants H to believe it too — that, to us, seems the essence of persuasion. The change allows persuasion to encompass situations where the dialogue is "evangelical" — where G wants to get other agents to agree with it because it feels so strongly that p is true and wants to broadcast the fact — as well as the situations that [20, 28] consider persuasions. Once again, the existing persuasion protocol from [20] will handle this weakening without alteration.

Table 5. An intermediate set of preconditions. $X \equiv B_G W_H p \wedge \neg B_G B_H p \wedge \neg B_G B_H \neg p$

		$B_G B_H p$	$B_G B_H \neg p$	$B_G \neg W_H p$	$B_G \neg W_H p$	X
				$B_G I_H A_{G,H} p$		
$B_G p$	$I_G B_H p$		Persuasion	Persuasion	Persuasion	Persuasion
$\neg W_G p$	$I_G W_G p$	InfoSeek	InfoSeek	Inquiry	Inquiry	InfoSeek
	$I_G W_G p$	Query	Query	Query**2**	Query**2**	Query
	$I_G B_G p$	Query**3**	Query**3**	Query**3**	Query**3**	Query**3**
	$I_G B_G B_H p$	Verify	Verify	Verify	Verify	Verify

Table 6. Our preconditions. $X \equiv B_G W_H p \wedge \neg B_G B_H p \wedge \neg B_G B_H \neg p$

		$B_G B_H p$	$B_G B_H \neg p$	$B_G \neg W_H p$	X
$B_G p$	$I_G B_H p$		Persuasion	Persuasion	Persuasion
$\neg W_G p$	$I_G W_G p$	InfoSeek	InfoSeek	Inquiry	InfoSeek
	$I_G W_G p$	Query	Query	Query**2**	Query
	$I_G B_G p$	Query**3**	Query**3**	Query**3**	Query**3**
	$I_G B_G B_H p$	Verify	Verify	Verify	Verify

In inquiry dialogues, it does not seem necessary for H to have the goal of establishing the truth of p. So long as one participant in an inquiry sets it off, all that is required of the other participant is that it respond truthfully and cooperatively during its turn, filling in missing pieces of the proof to the best of its ability. As a result, we drop the requirement $I_H A_{G,H} p$. The protocol for inquiry given above will work under this alteration to the preconditions since it makes no assumptions about H's goals.

Finally, from the perspective of G trying to decide what dialogues they can engage in under specific conditions, this considerably eases G's job since it no longer has to figure out what H's intentions are. The third and fourth columns of Table 5 thus collapse, and we are left with Table 6.

5 Conclusion and Future Work

This paper has considered dialogues about beliefs — that is dialogues akin to the ones that Walton and Krabbe [28] called information seeking, inquiry and persuasion — and, in particular, has systematically considered the preconditions for such dialogues. Doing so has exposed a need for a number of new kinds of dialogue (Verify, Query, Query**2** and Query**3**), and we have given protocols for these. Of course there is no more reason to think that this set of dialogues is complete, than there was any reason to suspect that the set originally identified by Walton and Krabbe was complete — the dialogues we have listed here, and the preconditions for them, just represent our current understanding.

It is useful to have identified these additional kinds of dialogues, which seem distinct from those proposed by Walton and Krabbe and commonly discussed in the literature. While the philosophical distinctions between these new types and the familiar ones – information seek, inquiry, and persuasion – are perhaps minor, the practical importance is major. These new dialogues are themselves useful — we started on this line of work because we identified the need for the Verify dialogue in the context of work on delegation — and if we are going to build agents that engage in these dialogues we need to identify protocols for them.

The desire to build agents that can engage in dialogue also explains why we have bothered to tease out the preconditions in such detail. As we have stressed throughout the paper, identifying which preconditions go with which dialogue (and hence with which protocol) is important so that an agent can choose which protocol it should make use of depending on what it knows about the agent with which it proposes to converse. Thus we see the preconditions as a necessary step towards operationalizing dialogue, and the statement of the preconditions in terms of mental notions (which Walton and Krabbe were largely careful to skirt) is a necessary step in doing this.

Having identified these new forms of dialogue, we need to examine their properties, just as [22] did for persuasion, information seeking and inquiry dialogues. We also plan to continue our analysis of dialogues about actions, that is, to expand into the territory of the kinds of dialogue that Walton and Krabbe called deliberation and negotiation.

As the different forms of dialogue multiply, it seems increasingly likely that we will not directly program agents with a range of different protocols of the kind described in this paper. Instead, we will program agents with the kinds of *atomic* protocols discussed in [18] — sub-protocols from which more complex protocols can be constructed. These atomic protocols will then be used to construct the kinds of protocol described here, enabling agents to verify, query, persuade, inquire, and information seek. However, in order to do this, we need to develop rules for composing atomic protocols to build up a range of complex interactions, and how to do this is a topic of our ongoing work.

Acknowledgements. This work was made possible by funding from NSF #REC-02-19347, NSF #IIS 0329037, EU FP6-IST 002307 (ASPIC), PSC-CUNY Award #66171-00 35 and PSC-CUNY Award #67437-00 36. The authors are grateful to the anonymous referees for their constructive comments and to Rohit Parikh for a comment that led to one of our examples.

References

[1] L. Amgoud. *Contribution a l'integration des préferences dans le raisonnement argumentatif.* PhD thesis, Université Paul Sabatier, Toulouse, July 1999.

[2] L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation framework. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 1–7, 1998.

[3] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In E. Durfee, editor, *Proceedings of the Fourth International Conference on Multi-Agent Systems*, pages 31–38, Boston, MA, USA, 2000. IEEE Press.

[4] T. J. M. Bench-Capon, F. P. Coenen, and P. Orton. Argument-based explanation of the British Nationality Act as a logic program. *Computers, Law and AI*, 2(1):53–66, 1993.

[5] Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert der van Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16, Montreal, Canada, 2001. ACM Press.

[6] F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In C. Castelfranchi and Y. Lespérance, editors, *Seventh Workshop on Agent Theories, Architectures, and Languages*, pages 141–156, Boston, USA, 2000.

[7] 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.

[8] P. E. Dunne. Prevarication in dispute protocols. In G. Sartor, editor, *Proceedings of the Ninth International Conference on AI and Law (ICAIL-03)*, pages 12–21, New York, NY, USA, 2003. ACM Press.

[9] D. M. Gabbay and J. Woods. More on non-cooperation in Dialogue Logic. *Logic Journal of the IGPL*, 9(2):321–339, 2001.

[10] D. M. Gabbay and J. Woods. Non-cooperation in Dialogue Logic. *Synthese*, 127(1-2): 161–186, 2001.

[11] R. Girle. Commands in Dialogue Logic. In D. M. Gabbay and H. J. Ohlbach, editors, *Practical Reasoning: Proceedings of the First International Conference on Formal and Applied Practical Reasoning (FAPR 1996), Bonn, Germany*, Lecture Notes in Artificial Intelligence 1085, pages 246–260, Berlin, Germany, 1996. Springer.

[12] T. F. Gordon. The Pleadings Game: An exercise in computational dialectics. *Artificial Intelligence and Law*, 2:239–292, 1994.

[13] K. Greenwood, T. Bench-Capon, and P. McBurney. Structuring dialogue between the People and their representatives. In R. Traunmüller, editor, *Electronic Government: Proceedings of the Second International Conference (EGOV03), Prague, Czech Republic*, Lecture Notes in Computer Science 2739, pages 55–62, Berlin, Germany, 2003. Springer.

[14] C. L. Hamblin. *Fallacies*. Methuen and Co Ltd, London, UK, 1970.

[15] J. Hulstijn. *Dialogue Models for Inquiry and Transaction*. PhD thesis, Universiteit Twente, Enschede, The Netherlands, 2000.

[16] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[17] P. McBurney and S. Parsons. Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and Artificial Intelligence*, 32(1–4):125–169, 2001.

[18] S. Parsons, P. McBurney, and M. Wooldridge. The mechanics of some formal inter-agent dialogue. In F. Dignum, editor, *Advances in Agent Communication*. Springer-Verlag, Berlin, Germany, 2003.

[19] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261—292, 1998.

[20] S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *1st International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 2002.

[21] S. Parsons, M. Wooldridge, and L. Amgoud. On the outcomes of formal inter-agent dialogues. In *2nd International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 2003.

[22] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[23] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.

[24] C. Reed. Dialogue frames in agent communications. In Y. Demazeau, editor, *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 246–253. IEEE Press, 1998.

[25] E. Sklar and S. Parsons. Towards the application of argumentation-based dialogues for education. In C. Sierra and E. Sonenberg, editors, *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems*. IEEE Press, 2004.

[26] K. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the Eleventh Joint Conference on Artificial Intelligence*, pages 517–523, 1989.

[27] K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.

[28] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, USA, 1995.

[29] T. Yuan, D. Moore, and A. Grierson. Educational human-computer debate: A computational dialectics approach. In G. Carenini, F. Grasso, and C. Reed, editors, *Proceedings of the Workshop on Computational Models of Natural Argument*, 2002.

Argumentation Based Modelling of Embedded Agent Dialogues

Yannis Dimopoulos[1], Antonis C. Kakas[1], and Pavlos Moraitis[2]

[1] Department of Computer Science, University of Cyprus
P.O. Box 20537, CY1678, Nicosia, Cyprus
{yannis, antonis}@cs.ucy.ac.cy
[2] Department of Mathematics and Computer Science
University René Descartes
45 rue des Saints-Pères, 75270 Paris Cedex 06, France
pavlos@math-info.univ-paris5.fr

Abstract. This paper presents a novel approach to modelling embedded agent dialogues. It proposes a specific structure for the supporting information accompanying the arguments that agents exchange during a dialogue, it defines formally how this information relates to the agent theory, and assigns to it semantics that is associated to each of the atomic dialogue types of the Walton-Krabbe typology. This allows the formal definition of necessary and sufficient initiation and acceptance conditions of licit dialectical shifts that are necessary for the modelling of embedded agent dialogues.

1 Introduction

The task of modelling agent dialogues has proved to be of great importance in representing complex agent interactions. Since the work of Walton and Krabbe [16] proposing a classification of possible atomic dialogue types (i.e. deliberation, negotiation, persuasion, information-inquiry, information-seeking, eristic) a lot of work have been devoted to modelling the first five of them, the sixth being considered unappropriate in a multi-agent context. Recently, some of this work has adopted an argumentation-based approach for such dialogue modelling as can be found for example in [13], [1], [14], [6], [11], [12]. However, to our knowledge, there exists only a few cases (see e.g. [10], [8], [14]) of study of the combination of atomic dialogues and of the particular combination of embedded dialogues.

Embedded dialogues are a very interesting combination of atomic dialogues. They concern situations where during a specific dialogue type, the interlocutors can shift to another dialogue type. When this subsidiary dialogue closes a shift back is made to the external dialogue which will continue from the point where it was interrupted. As Walton [15] says: "the one dialogue can be "sandwiched in" between the prior and subsequent parts of an enveloping sequence of dialogue of another type. Practical reasons can cause the interruption, but then the dialogue can quickly shift back to the original type". In the case of embedded dialogues the outcome of the second dialogue can influence the quality of the outcome of

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 169–181, 2006.

the original dialogue, because the second dialogue is functionally related to the argumentation in the first dialogue.

An important issue in the multi-agent context, is related to the ability of detecting in a current dialogue, licit dialectical shifts, which according to the literature (see e.g. [15]), are those that allow agents to transit to another type of dialogue which supports the old goals or at least allows their fulfilment to be carried forward. Such dialogues shifts to embedded dialogues are useful in contributing to the successful completion of the outer dialogues. If the new dialogue is blocking the old goals, the dialectical shift is considered illicit and it is often associated with informal fallacies [15] which we believe are less appropriate for artificial agents dialogues.

In this paper we investigate how to model embedded dialogues based on the argumentation reasoning of the agent. We present an argumentation framework in which an agent represents and reasons with the various components of its knowledge and dialogue theory. Based on this the agent is equipped with a set of different capabilities for reasoning about goals, beliefs and actions. We then define formally the structure of the supporting information accompanying the exchanged arguments between the agents during a dialogue and present how its constituents are related to goals, beliefs and actions. This allows us to link the argumentation-based reasoning of the agent to its dialogues and formalize within the framework the five atomic dialogue types of the Walton-Krabbe typology. In turn we can give a formal notion of licit dialectical shifts (in the context of embedded dialogues) through the definition of initiation conditions for the five atomic dialogue types and acceptance conditions for such dialectical shifts. To our knowledge, our work is one of the first attempts to provide formal definitions for all these issues related to the modelling of embedded dialogues.

The rest of the paper is organized as follows. Section 2 presents briefly the underlying argumentation theory and the primitive components that a framework needs to possess in order to build embedded dialogues. Section 3 defines the dialogue supporting information while section 4 presents the embedded dialogue framework we propose. Finally, section 5 discusses related work and concludes.

2 Background

2.1 Basic Argumentation Theory

This section gives briefly the basic concepts of the underlying argumentation framework in which an agent represents and reasons with its communication theory. With this the agent will be able to generate, and then communicate to other agents, different arguments for the various topics involved in its dialogues. There are two important features of an argumentation framework that are required for this purpose. Firstly, the framework needs to be *adaptive* to changes in the current knowledge of the agent about the state of the world. Secondly, the framework should be able to identify in its arguments a set of (significant) conditions on which an argument is *supported.* In particular, this set may contain assumptions pertaining to the incomplete information that the agent has about the world. Any argumentation framework that can provide these two functions is suitable.

An argumentation framework in its abstract form is based on a set, $\mathcal{A}$, of arguments and a binary attacking relation, $\mathcal{AR}$, amongst these arguments. We will assume that arguments in $\mathcal{A}$ are represented by logical theories in some background monotonic logic whose derivability relation we will denote by $\vdash_B$. Each argument A is a subset of a given theory $\mathcal{T}$ and we say that A is an *argument for* L whenever $A \vdash_B L$. An example of such a framework, called Logic Programming without Negation as Failure ($LPwNF$), was proposed in [5] in which theories are written in terms of Extended Logic Programming rules and priorities on these rules. This was developed further in [7] providing the above desired features of adaptability and supportedness of arguments. Although not crucial for the work in this paper we will adopt this framework in order to be more concrete in our presentation.

In this framework of $LPwNF$ the attacking relation $\mathcal{AR}$ is realized via a (symmetric) notion of *incompatibility* between literals, that defines when two literals cannot hold together, and a set of priority rules, given within the same theory $\mathcal{T}$. Informally, given two subsets A', A of $\mathcal{T}$, A' attacks A if they have incompatible consequences under the background logic $\vdash_B$ and A' is stronger than A according to the priority rules in the theory. Thus a given argumentation theory $\mathcal{T}$ defines both the set of arguments and the attacking relation amongst them.

The central notion for the acceptance of an argument is that of *admissibility.* This and the argumentation entailments that follow from it are defined as follows.

Definition 1. *Let $\mathcal{T}$ be an argumentation theory. An argument $\Delta \subseteq \mathcal{T}$ is* admissible *iff Δ does not attack itself (it is consistent) and for any $\Delta' \subseteq \mathcal{T}$ if Δ' attacks Δ then Δ attacks Δ'.*

Given a literal L then L is a skeptical consequence *of the theory iff L holds, under the background monotonic logic $\vdash_B$, in an admissible subset of $\mathcal{T}$ and for any literal, $\overline{L}$, which is incompatible with L, there exists no admissible argument in which $\overline{L}$ holds under $\vdash_B$.*

In several cases we want to base the admissibility of an argument on some significant information about the specific case in which we are reasoning or on incomplete information that is missing from our theory. We can formalize this conditional form of argumentative reasoning by defining the notion of *supporting information* and extending argumentation with abduction on this information.

Definition 2. *Let $\mathcal{T}$ be an argumentation theory and, Ab, a distinguished set of predicates in the language of $\mathcal{T}$, called* abducible *predicates. Given a literal L, a* supported argument *for L is a tuple (Δ, S), where S is a set of ground abducible facts not in Δ such that Δ is not an admissible argument for L, but $\Delta \cup S$ is an admissible argument for L. We say that S is supporting information for the argument Δ of L.*

Given this we have an argumentation entailment, $\vdash_{arg}$, defined as follows.

Definition 3. *Let $\mathcal{T}$ be an argumentation theory and, Ab, a distinguished set of* abducible *predicates. Given a literal L, $T \vdash_{arg} L$, iff there exists a set of ground abducible facts S such that L is a skeptical consequence of $\mathcal{T} \cup S$. In other words, there exists in $\mathcal{T}$ a supported argument (Δ, S) for L and for any literal $\overline{L}$ which is incompatible with L there exists no supported argument for $\overline{L}$ in $\mathcal{T} \cup S$.*

2.2 Primitives for Embedded Dialogues

In this section we present the primitives components that a framework needs to possess in order to build embedded dialogues within this framework.

Reasoning Capabilities of an Agent. Dialogues depend on the reasoning capabilities of the agents. We consider that different reasoning capabilities are involved in the reasoning process of the agents, during the different possible types of dialogues. This reasoning process may concern a goal decision reasoning capability for the choice of the preferred goal to be achieved, a temporal reasoning capability about actions and change for deriving its beliefs about the current (or future state of the work) and a plan preference capability for deriving preferred plans for a goal. In this paper we consider that all these different capabilities can be derived via suitable argumentation theories in the argumentation framework described above. The importance of using argumentation as a basis for the reasoning capabilities stems from the fact that agents can then exchange, during their dialogue, their arguments (and the supporting information for these) and use these to develop their dialogues.

We will assume that agents have the following argumentation based capabilities that operate on their knowledge T:

- a preferred plan capability, $\vdash_{PPlan}$, which is given by the synthesis of planning capability; $\vdash_{Plan}$, and a plan preference selection,$\vdash_{PP}$. Hence given a goal G, if $T \vdash_{PPlan} plan(G)$ then $plan(G)$ is a preferred plan for the goal G. We will also write $T \vdash_{PPlan} G$ to mean that there exists a preferred plan for the goal G.
- a desired goal capability, $\vdash$, that derives goals which are currently preferred by the agent and for which it also has a (preferred) plan to satisfy. Hence, $T \vdash G$ can be decomposed into a *goal decision* capability, $T \vdash_{GP} G$, and the $T \vdash_{PPlan} G$.
- a temporal reasoning or Reasoning about Actions and Change (RAC) capability, $\vdash_{RAC}$, with which the agent is able to derive its beliefs about the current (or future) state of the world. This is based on the agent's knowledge of action effect laws (and constraints) and on narrative knowledge about the past containing actions that have occurred and past observations of properties of the world.

Dialogue Supporting Information. The supporting information accompanying the arguments the agents exchange during a dialogue is important for various reasons. For example, we will see that it helps characterize the type of the appropriate dialogue to be undertaken according to the topic to be discussed at a certain instant of a specific dialogue type. In this paper we will structure the supporting information according to how it relates to the goals, beliefs and actions of the agents. Supporting information comes from the underlying argumentation reasoning (as described above) that is used to implement the reasoning capabilities of the agent.

For the planning capability $\vdash_{PPlan}$ any generated plan, $plan(G)$, itself forms supporting information in the form of future actions for the arguments that derive the goal G. (Note that part of the plan can be requests for other agents to achieve a needed subgoal for G.) For the desired goal capability $\vdash$ an admissible argument will contain in its support conditions for the goal to be both desired and have a preferred plan (intention) under which the agent aims to achieve it. Finally, for the temporal reasoning capability the support, S, of arguments for current beliefs contains assumptions on properties at earlier times which are unknown to the agent and therefore it needs to hypothesize these.

We will see below that (part of) this support will be communicated with the aim to inform the other agent the *Reasons* why the agent wishes to achieve the goal G in this way and the *Terms* that the agent requires from the other agent in its endeavor for G.

Atomic Dialogues Initiation Conditions. A specific dialogue type can be initiated only under certain necessary conditions. We will propose a formal definition of such initiation conditions for the five atomic dialogues types of the Walton-Krabbe typology, based on a synthesis of informal descriptions proposed in the literature, but we do not pretend that these conditions may cover the totality of the possible situations. The definition of these initiation conditions will be based on arguments of the agents for their desired goals and their supporting information.

Dialectical Shift. A dialectical shift (see e.g [15]) is a transit from a certain type of dialogue to another of different type. This transit might allow agents to achieve goals whose fulfillment was impossible in the originally open dialogue. The definition of such a dialectical shift corresponds to a set of sufficient conditions under which such a transit is possible. The acceptance conditions of such a shift must also be defined. Our definitions for these will again be based on the arguments and supporting information exchanged during the dialogue so far. Here again, we will not claim that our formalization is complete but rather that it forms a core that can be extended as needed for increasingly complex situations.

3 Dialogue Supporting Information

Agents operate in a dynamic and ever changing world. To keep track of the change an agent uses his capability, $\vdash_{RAC}$, to derive conclusions about how the world is in its current state (to simplify our discussion we assume that an agent never needs to reason about the past). We call *current atoms* (literals) the atoms (literals) of the theory of the agent that refer to the current state of the world. A current literal p is called a *belief* if $T \vdash_{RAC} p$.

An agent can execute *actions* that can change the current state of the world to some other more "desirable" state. This new state is described via the set of *goals* that the agent wishes to achieve through the execution of actions. Atoms (literals) that refer to some future state of the world are called *future atoms* (literals). A goal G is a conjunction or set of future literals some of which are not true in the current state of the world, such that $T \vdash G$.

We will call *locutions* or *dialogue moves* the sentences that are exchanged between the agents during a dialogue. Locutions are 4-tuples of the form $P(a, b, t, Content)$ where P is a *performative* contained in a set that is in the lines of those used in [2], a is the agent that utters the locution, b is the intended recipient of the locution and t specifies the type of the current dialogue $\mathcal{D}$ the locution is uttered or the type of the dialogue to be initiated by the current locution. The $Content$ of the message is a 3-tuple of the form $< topic, reason, terms >$ where *topic* concerns the subject of the specific dialogue and it may be a set of goals, beliefs or actions of the involved agents and the other, possibly empty, fields correspond to the *supporting information* of the argument proving the literals contained in the field topic.

The proposed structure for the supporting information is partially inspired by the work presented in [11]. The literals that appear in the set *reason* refer to what the agent believes is true in the current state of the world, whereas the literals in the *terms* refer to what must be true in the future so that his goals succeed. More specifically, the set *terms* is the union of two subsets TR^- and TR^+ with the following meaning. If $p \in TR^-$, then for any other agent β it must be the case that $T_\beta \not\vdash \neg p$ whereas if $p \in TR^+$ then for some other agent β it must be the case that $T_\beta \vdash p$. Intuitively, the literals in TR^- refer to actions or goals that the other agents should refrain from executing or achieving, whereas the literals in TR^+ refer to actions or future literals that the agent requests that other agents will execute or achieve.

In a similar way, the set *reasons* of an agent α is divided in two subsets, R^K and R^U. The set R^K contains a current literal p iff $T_\alpha \vdash_{RAC} p$, whereas R^U contains current literals that are assumptions made by agent α. By placing a current literal p in the set R^U of a locution, an agent declares that he assumes that p has the value true as he has no sufficient information from which he can derive the value of p. Therefore, the content of a locution is a 3-tuple of the form $< TP, < R^K, R^U >, < TR^+, TR^- >>$, where TP is the topic as noted above.

In this paper we assume that the agents are truthful, in the sense that the information they communicate with other agents is a consequence of their knowledge bases. Formally, if $P(a, b, t, < TP, R, TR >)$ is a locution sent by agent a to agent b it must be the case that the theory T_a of agent a has an admissible argument (Δ_a, S_a) such that $(\Delta_a, S_a) \vdash TP$ and $R \cup TR \subseteq S_a$.

4 The Embedded Dialogue Framework

In this section we present formally our framework for embedded dialogues. We will restrict our attention to dialogues between two agents. In this context a dialogue is defined as follows.

Definition 4. *Dialogue*
A dialogue $\mathcal{D}$ between agents α and β is a finite sequence of the form $\mathcal{D}$=$L_1^{k|l} L_2^{l|k} \dots L_m^{j|n}$ with $k, l \in \{\alpha, \beta\}$, where each element $L_i^{p|q}$, called the i dialogue step, *is a locution of the form $P(p, q, t, C)$, and $j = k$, $n = l$ if m is odd and $j = l$, $n = k$ if m is even.*

We define now the outcome of a dialogue, and its sub-dialogues, for each of the participating agents. This definition is in line with the one presented in [13].

Definition 5. *Dialogue Outcome*
Let $\mathcal{D}$=$L_1^{k|l} L_2^{l|k} \dots L_m^{j|n}$ be a dialogue between agents α and β, with $L_i^{p|q} = P^i(p, q, t, < TP_i, < R_i^K, R_i^U >, < TR_i^+, TR_i^- >>)$. The outcome of $\mathcal{D}$ for agent α is defined as the set $O_{\mathcal{D}}^{\alpha} = \bigcup_{i=1}^{m} \{s | s \in TP_i \cup R_i^K \cup TR_i^-$, for $L_i^{\alpha|\beta} \in \mathcal{D}$ and $P^i = accept\}$. Similarly, the outcome of $\mathcal{D}$ for agent β is the set $O_{\mathcal{D}}^{\beta} = \bigcup_{i=1}^{m} \{s | s \in TP_i \cup R_i^K \cup TR_i^-$, for $L_i^{\beta|\alpha} \in \mathcal{D}$ and $P^i = accept\}$.

Given a dialogue $\mathcal{D}$=$L_1^{k|l} L_2^{l|k} \dots L_r^{c|d} \dots L_m^{j|n}$ between agents α and β, $O_{\mathcal{D}_r}^{\alpha}$ denotes the outcome for agent α of the sub-dialogue that starts at step 1 and ends at step r, and is defined as $O_{\mathcal{D}_r}^{\alpha} = O_{\mathcal{D}'}^{\alpha}$, where $\mathcal{D}'$ is the dialogue $\mathcal{D}' = L_1^{k|l} L_2^{l|k} \dots L_r^{c|d}$. The definition of $O_{\mathcal{D}_r}^{\beta}$ is similar.

The theory of an agent, and therefore his beliefs, goals and plans, change during the course of a dialogue. These changes are realized via the function $rev(T, S)$ that takes a theory T and a set of literals S and revises T to a new theory T' so that $T' \vdash s$ for all $s \in S$.

Definition 6. *Agent Theories and Agent Goals*
If $\mathcal{D}$ is a dialogue between agents α and β, $T_{\alpha}^{\mathcal{D}_i}$ denotes the theory of agent α at step i of the dialogue $\mathcal{D}$, and is defined as $T_{\alpha}^{\mathcal{D}_i} = rev(T_{\alpha}, O_{\mathcal{D}_i}^{\alpha})$, where T_{α} is the theory of agent α at the beginning of the dialogue.
The goal of agent α at step i of dialogue $\mathcal{D}$ is denoted by G_{α}^{i} and is a set of future literals such that $T_{\alpha}^{\mathcal{D}_i} \vdash G_{\alpha}^{i}$.

4.1 Modelling Dialectical Shifts

In this subsection we present formal definitions for the initiation conditions of the five dialogue types of the Walton-Krabbe typology, the notion of licit dialectical shift [15], the acceptance conditions of such a shift and the notion of efficient dialectical shift. These definitions aim to capture informal descriptions, commonly accepted in the literature. The initiation conditions allow an agent to detect the possibility of a shift from the current dialogue to another dialogue of a different type. They are necessary conditions for a dialogue shift to occur. The initiation conditions are linked to the constituents of the content of the locutions exchanged between agents, which correspond to the supporting information of the arguments used by the agents during a dialogue.

A dialectical shift from a dialogue of any type different than negotiation to a negotiation dialogue means that either the participating agents have conflicting goals (or interests) (see e.g. [11]) or the terms in the locution of one of the agents leads to the failure of the goals of the other agent. This is a more general consideration for negotiation than the one proposed in the Walton and Krabbe typology where negotiation concerns the division of some scarce resource. Formally, this type of shift is defined as follows.

Definition 7. *Negotiation*
Let α and β be two agents involved in a dialogue $\mathcal{D} = L_1^{k|l} L_2^{l|k} \ldots L_i^{\beta|\alpha}$, with $L_i^{\beta|\alpha} = P^i(\beta, \alpha, t, < TP_i, R_i, TR_i >)$, and G_α^i the goal of agent α at step i of $\mathcal{D}$. Agent α can start a negotiation dialogue at step $i+1$ of $\mathcal{D}$ if either $\neg G_\alpha^i \in TP_i$, or for all admissible arguments $(\Delta_\alpha, S_\alpha)$ of theory $T_\alpha^{\mathcal{D}_i}$ such that $(\Delta_\alpha, S_\alpha) \vdash G_\alpha^i$ there is $L \in S_\alpha$ s.t. $\neg L \in TR_i$.

For the deliberation dialogue there is no obvious definition for the initiation conditions. However we tried to capture as much as possible the intuition proposed in the literature (see e.g. [4],[8],[9]). According to this definition the shift to a deliberation dialogue happens when the participants seeking to agree upon an action or a course of action which is needed in some circumstance. In order to give a formal definition, in this paper we make the assumption, that the action to be discussed contributes to the achievement of some goal of the participants, or to the achievement of a common goal.

Definition 8. *Deliberation*
Let α and β be two agents involved in a dialogue $\mathcal{D} = L_1^{k|l} L_2^{l|k} \ldots L_i^{\beta|\alpha}$. Agent α can start a deliberation dialogue on an action p, with $L_{i+1}^{\alpha|\beta} = P^{i+1}(\alpha, \beta, t, < TP_{i+1}, R_{i+1}, < TR_{i+1}^{+}, TR_{i+1}^{-} >>)$ with $p \in TP_{i+1}$, if $T_\alpha^{\mathcal{D}_i} \vdash_{GP} G$, $T_\alpha^{\mathcal{D}_i} \not\vdash_{Plan} G$, $T_\alpha^{\mathcal{D}_i} \cup p \vdash G$ and $T_\alpha^{\mathcal{D}_i} \not\vdash p$ and where G is a future literal.

The shift to a persuasion dialogue means that one agent disagrees with the beliefs of the other agent. The formal details are as follows. Currently in our work, persuasion is only concerned with the beliefs of agents. This is in line with the literature (see e.g. [1],[8]). However in some works persuasion is also concerned with actions. It is easy to see that a definition similar with the one proposed in the following could be proposed for the actions of agents.

Definition 9. *Persuasion*
Let α and β be two agents involved in a dialogue $\mathcal{D} = L_1^{k|l} L_2^{l|k} \ldots L_i^{\beta|\alpha}$, with $L_i^{\beta|\alpha} = P^i(\beta, \alpha, t, < TP_i, < R_i^K, R_i^U >, TR_i >)$. Agent α can start a persuasion dialogue at step $i+1$ of $\mathcal{D}$, if there exists a current literal p such that $T_\alpha^{\mathcal{D}_i} \vdash_{RAC} p$ and $\neg p \in TP_i \cup R_i^K$.

A shift to an information-inquiry dialogue is similar to the shift to a deliberation dialogue, their main difference being the former concerns current literals (i.e. beliefs) while the latter actions. Informally, a shift to an information-inquiry dialogue means that one of the agents can provide to the other, part of the proof of some current literal the truth-value of which is unknown to both. This is in line with the literature (se e.g. [1],[8]).

Definition 10. *Information-Inquiry*
Let α and β be two agents involved in a dialogue $\mathcal{D} = L_1^{k|l} L_2^{l|k} \dots L_i^{\beta|\alpha}$. Agent α can start an information-inquiry dialogue at step $i+1$ of $\mathcal{D}$, on a current literal s with $L_{i+1}^{\alpha|\beta} = P^{i+1}(\alpha, \beta, t, < TP_{i+1}, < R_{i+1}^K, R_{i+1}^U >, TR_{i+1} >)$ s.t. $s \in TP_{i+1}$ and another current literal $p \in R_{i+1}^U$, if $T_\alpha^{\mathcal{D}_i} \nvdash_{RAC} s$, $T_\alpha^{\mathcal{D}_i} \cup p \vdash_{RAC} s$ and $T_\alpha^{\mathcal{D}_i} \nvdash_{RAC} p$.

This definition means that the agent α will start an information-inquiry dialogue if he searches for the truth-value of a current literal s, he knows that it can be proven by using the truth-value of the current literal p but he cannot prove p. That is why he wants to start a dialogue with another agent β who is also interested in the truth-value of s, who cannot prove s, but he can prove p.

Finally, a shift to an information-seeking dialogue is possible only if the truth-value of some current literal is unknown to one agent but known to the other. This is also in line with the literature (se e.g. [1],[8]).

Definition 11. *Information-Seeking*
Let α and β be two agents involved in a dialogue $\mathcal{D} = L_1^{k|l} L_2^{l|k} \dots L_i^{\beta|\alpha}$. Agent α can initiate an information-seeking dialogue at step $i+1$ of $\mathcal{D}$, if there exists a current literal p s.t. $T_\alpha^{\mathcal{D}_i} \nvdash_{RAC} p$, $T_\alpha^{\mathcal{D}_i} \nvdash_{RAC} \neg p$.

According to Walton [15], a dialectical shift from one dialogue type to another is *licit* if it contributes to the fulfilment of the goals of the original dialogue. If the new dialogue appears to block these goals, this shift is considered *illicit* and it is often associated with informal fallacies which are inappropriate in artificial agents dialogues. Thus, in our framework we only consider the case of licit dialectical shifts and capture this property in the following definition.

Definition 12. *Licit dialectical shift*
Let α and β be two agents participating in a dialogue $\mathcal{D}$ of type t and G_β^i the goal of agent β at step i of $\mathcal{D}$. Furthermore, let $L_i^{\alpha|\beta} = P^i(\alpha, \beta, t, < TP_i, R_i, TR_i >)$ be the locution sent by agent α to agent β at step i of $\mathcal{D}$. Agent β will initiate an embedded dialogue $\mathcal{D}'$ of type $t' \neq t$ with dialogue topic TP_{new} s.t. $TP_{new} \subseteq TP_i \cup R_i \cup TR_i$ if the following conditions hold:

1) The initiation conditions of the dialogue type t' hold
2a) $T_\beta^{\mathcal{D}_i} \cup O_{\mathcal{D}_i}^\alpha - K \nvdash g_\beta^i$ for all $g_\beta^i \subseteq G_\beta^i$, and $K = \{p | \neg p \in R_i \cup TR_i\}$ if $t' \in \{negotiation, persuasion\}$.
2b) $T_\beta^{\mathcal{D}_i} \cup O_{\mathcal{D}_i}^\alpha \nvdash g_\beta^i$ for all $g_\beta^i \subseteq G_\beta^i$ if $t' = deliberation$
3) $T_\beta^{\mathcal{D}_i} \cup TP_{new} \vdash g_\beta^i$ for some $g_\beta^i \subseteq G_\beta^i$
4) $P(\beta, \alpha, t, < TP_{new}, R_{new}, TR_{new} >)$ is not a legal locution for all possible R_{new}, TR_{new}, and $P(\beta, \alpha, t', < TP_{new}, R_{new}, TR_{new} >)$ is a legal locution for all possible R_{new}, TR_{new}.

Informally this definition says that the agent β will initiate an embedded dialogue $\mathcal{D}'$ of type t' on a new topic TP_{new} if:

1) The initiation conditions of the dialogue type t' hold
2) The agent cannot prove any goal if he removes from his knowledge the literal p whose negation belongs either to the reason or to the terms of the received locution. In the former case p can be a belief and the new dialogue will be a persuasion dialogue while in the later p can be an action or goal and the new dialogue will be a negotiation one.
3) With the new topic the agent β will be able to prove some goal
4) The locution with the new topic is not a legal locution in the current dialogue type but it is a legal locution in the new dialogue type.

Here we note that the definition of the legality of a locution depends on the adopted dialogue framework and its protocols but the exact details are beyond the scope of this paper.

Above we have not considered shifts to *information-seeking* or *information-inquiry* dialogues. For these two types of dialogues we will assume that their initiation conditions are in fact the necessary and sufficient conditions of a licit dialectical shift to them.

Finally, we define the criteria under which an agent participating in a dialogue $\mathcal{D}$ of type t accepts the request of his interlocutor to enter a new (embedded) dialogue of type t' in order to continue their discussion.

Definition 13. *Dialectical shift acceptance*
Let $\mathcal{D}$ be an open dialogue of type t between two agents α and β, and G^i_β the goal of agent β at step i of t. Furthermore, let $L^{\alpha|\beta}_i = P^i(\alpha, \beta, t', < TP_i, R_i, TR_i >)$ be the locution sent by agent α to agent β at step i of the current dialogue $\mathcal{D}$ in order to initiate an embedded dialogue $\mathcal{D}'$ of type $t' \neq t$. Agent β has to accept entering the new dialogue if the following conditions hold:

1) The initiation conditions of the dialogue type t' hold with $t' \in \{negotiation, persuasion, deliberation\}$
2) $T^{\mathcal{D}_i}_\beta \cup O^\alpha_{\mathcal{D}_i} \nvdash g^i_\beta$ for all $g^i_\beta \subseteq G^i_\beta$
3) $TP_{new} \subseteq TP_{i-1} \cup R_{i-1} \cup TR_{i-1}$ holds for the locution $L^{\beta|\alpha}_{i-1} = P^{i-1}(\beta, \alpha, t, < TP_{i-1}, R_{i-1}, TR_{i-1} >)$ sent at step $i-1$ of $\mathcal{D}$ by agent β to agent α and where $G^{i-1}_\beta \subseteq TP_{i-1}$

In the current stage of our work we consider that a dialectical shift to an *information-seeking* or *information-inquiry dialogue* is always accepted.

Our work concerns embedded dialogues among artificial agents. In this context an agreement is desirable and therefore our framework enforces agents to stay in a dialogue as long as possible by exploiting the possibility to shift among different types of dialogues according to the subject to be discussed. This is captured in condition 3 of the above definition which implies that agent β is obliged to accept a dialectical shift proposed by agent α if the proposed new topic is related to the topic, reasons or terms of the locution sent in the previous step by himself to agent α. However, one could remove some of the above conditions or add new ones depending on the context the embedded dialogue is taking place.

The rational behind the conditions in the definitions of the licit dialectical shift and the dialectical shift acceptance, is that an agent initiates or accepts the initiation of a new type of dialogue if he expects that the outcome of the new dialogue (in case it terminates successfully) will allow the achievement of a goal which is impossible in the current dialogue. This notion of dialectical shift *efficiency* is captured formally in the following definition.

Definition 14. *Efficient dialectical shift*
Let α and β be two agents participating in a dialogue $\mathcal{D}$ of type t and G^i_α and let G^i_β be the goals of the agents at step i of $\mathcal{D}$. An embedded dialectical shift to another dialogue type $\mathcal{D}'$ will be efficient for both agents iff the following conditions hold:

1) $T^{\mathcal{D}_i}_\alpha \nvdash g^i_\alpha$ for any $g^i_\alpha \subseteq G^i_\alpha$ and $T^{\mathcal{D}_i}_\alpha \cup O^\beta_{\mathcal{D}'} \vdash g^i_\alpha$ for some $g^i_\alpha \subseteq G^i_\alpha$
2) $T^{\mathcal{D}_i}_\beta \nvdash g^i_\beta$ for any $g^i_\beta \subseteq G^i_\beta$ and $T^{\mathcal{D}_i}_\beta \cup O^\alpha_{\mathcal{D}'} \vdash g^i_\beta$ for some $g^i_\beta \subseteq G^i_\beta$

The next proposition shows that if the conditions of a licit dialectical shift hold for one of the agents, and the acceptance conditions hold for the other, the shift to the new dialogue will lead to the achievement of both agents' goals.

Proposition 1. *During an atomic dialogue $\mathcal{D}$ of type $t \in \{deliberation, negotiation\}$ between two agents, if a licit dialectical shift to another atomic dialogue $\mathcal{D}'$ of type $t' \in \{deliberation, negotiation, persuasion\}$ with $t \neq t'$ is initiated by one of the agents and accepted by the other, and the dialogue $\mathcal{D}'$ terminates successfully then it is efficient for both.*

The following proposition is a direct consequence of the way the content of a locution is defined and related to the agent's theory.

Proposition 2. *During a persuasion dialogue $\mathcal{D}$ between two agents, a licit dialectical shift to another atomic dialogue $\mathcal{D}'$ of type t' with $t' \in \{deliberation, negotiation\}$ is not possible.*

This property illustrates the fact that in the current stage of our work a persuasion dialogue can only concern the beliefs of the agents and therefore a shift to a deliberation or negotiation dialogue which may concern actions or goals is not possible.

5 Related Work and Conclusions

In this paper we have presented a novel approach for modelling embedded agent dialogues. Although there exists some work on the combination of atomic dialogues (see e.g. [10], [8], [14]) none of this is completely devoted to the particular study of embedded dialogues. In our work we have laid out a formal framework based on the underlying argumentation reasoning of agents for the various issues which are necessary for the modelling of such dialogues. We have proposed a particular structure for the supporting information of the arguments exchanged between agents during a dialogue that is used to prompt (and facilitate) shifts from one dialogue type to another. We have defined the initiation conditions of the five atomic dialogues of the Walton-Krabbe typology, adopted in multi-agent context, and have shown how these are related to the supporting information. These definitions are based on a synthesis of informal descriptions proposed in the literature, but we do not pretend that these conditions may cover the totality of the possible situations. Some other works have also discussed initial conditions for the three of the five atomic dialogues (see e.g. [1], [8]) but only in a very abstract way and no direction has been given on how they could be used in the context of embedded dialogues.

Within our framework we have proposed a formal definition for the notion of licit dialectical shifts which is fundamental for the modelling of embedded dialogues along with acceptance conditions for such shifts for the participating agents. The allowed licit dialectical shifts in our framework are consistent with those of [16]. In [8] the authors have also proposed a formal framework for different atomic dialogues and have discussed issues on possible combinations. However the embedded dialogues are considered only as a case of combination of atomic dialogues with little particular attention on the formal definition of the special characteristics of such dialogues. Finally, we note that our dialogue theories for atomic and embedded dialogues can be easily implemented directly from their declarative specification in the Gorgias system [3] for argumentation and abduction.

The work presented in this paper is a first step to the formal study of embedded dialogues. Future work will concentrate on a more detailed investigation of the properties of our framework.

Acknowledgments. This work was partially supported by the IST programme of the EC, FET under the IST- 2001-32530 SOCS project, within the Global Computing proactive initiative.

References

1. L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of ICMAS98*, 1998.
2. L. Amgoud and S. Parsons. Agent dialogue with conflicting preferences. In *Proceedings of ATAL01*, 2001.
3. GORGIAS. A system for argumentation and abduction. http://www.cs.ucy.ac.cy/˜ nkd/gorgias, 2002.
4. D. Hitchcock, P. McBurney, and S. Parsons. A framework for deliberation dialogues. In *Proceedings of OSSA01*, 2001.
5. A. Kakas, P. Mancarella, and P. M. Dung. The acceptability semantics for logic programs. In *Proceedings of the International Conference on Logic Programming*, 1994.
6. A. Kakas, N. Maudet, and P. Moraitis. Layered strategies and protocols for argumentation-based agent interaction. In *Proceedings of ArgMAS04*, 2004.
7. A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of AAMAS03*, 2003.

8. P. McBurney and S. Parsons. Games that agents play: a formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11:315–334, 2002.
9. P. McBurney and S. Parsons. A denotational semantics for deliberation dialogues. In *Proceedings of AAMAS04*, 2004.
10. S. Parsons, P. McBurney, and M. Wooldridge. The mechanisms of some formal inter-agent dialogue. In F.Dignum, editor, *Proceedings of Advances in Agent Communication*. Springer-Verlag, 2003.
11. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3), 1998.
12. S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *Proceedings of AAMAS02*, 2002.
13. S. Parsons, M. Wooldridge, and L. Amgoud. On the outcomes of formal inter-agent dialogues. In *Proceedings of AAMAS03*, 2003.
14. C. Reed. Dialogue frames in agent communication. In *Proceedings of ICMAS98*, 1998.
15. D. N. Walton. Types of dialogues, dialectical shifts and fallacies. In *Argumentation Illuminated*. 1992.
16. D.N. Walton and E.C.W. Krabbe. *Commitment in dialogue: basic concepts of interpersonal reasoning*. State University of New York Press, 1995.

Liberalizing Protocols for Argumentation in Multi-agent Systems

Gerard A.W. Vreeswijk

Dept. of Computer Science
Utrecht University, The Netherlands
gv@cs.uu.nl

Abstract. This paper proposes a liberalized version of existing truth-finding protocols for argumentation, such as the standard two-agent immediate-response protocol for computing the credulous acceptance of conclusions in an argument system. In the new setup agents decide autonomously which issues need to be discussed, when to query other agents, when to keep on querying other agents, and when to settle for an answer. In this way, inter-agent disputes are regulated by the agents themselves, rather than by following an outlined protocol. The paper concludes with a prototype implementation and with a comparison of related work on conversation analysis and computational dialectic.[1]

1 Introduction

Argumentation has become increasingly important in multi-agent system (MAS) research. Modern MAS models require that agents are able to argue, for example to support their position in a negotiation or to explain a possibly controversial decision.

A great deal of research on defeasible reasoning and formal argumentation has been done in the past few years, and also a great deal of research on inter-agent inquiry dialogue has been accomplished. However, most research on argumentation in AI is devoted to monological (single-agent) algorithms and dialogical two-party immediate response dialectics that are sound and complete with respect to a particular argument semantics. Examples of such semantics are the grounded extension semantics, the stable extension semantics and the preferred extension semantics [8, 32]. Research on inter-agent inquiry, on the other hand, is concerned with studying sequences of conversation at the speech act level that are useful, orderly, effective [11, 25, 28] and sufficiently controllable by the agents that use them [2].

A remarkable difference between the two approaches is that argumentation dialogues are often extremely constrained and deterministic, while inter-agent inquiry dialogues are less constrained but also less concerned with getting the underlying argument semantics right [31]. Recently, a number of proposals have

[1] A (colorful and instructive) poster based on a shorter version of this paper was presented at AAMAS'05 [34]. The poster itself can be viewed at
`http://www.cs.uu.nl/~gv/abstracts/liberal_protocol_poster.pdf`.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 182–198, 2006.

been made to connect the two approaches, for example by dropping protocol constraints [23] or, conversely, by formulating desiderata for argumentation protocols [15, 17].

This paper proposes a minimalistic but complete model for inter-agent argumentation that is less constrained than existing argumentation protocols. The model is minimalistic in the sense that the agent architecture, the internal knowledge representational language and the message format are minimalistic and contain just enough detail to "keep the agents going". The model is complete in the sense that it describes the entire setup—from agent internals to communication language—and possesses enough detail to obtain a runnable MAS.

The purpose of the model is give the minimal means with which agents can engage in a dispute that is brought about by the agents themselves (autonomous agent perspective) rather than that the agents follow a fixed and external protocol (defeasible argumentation perspective). The resulting system is suitable for parametrization, experimentation and analysis.

The paper is structured as follows. In Sec. 2 the global setup is described. Sec. 3 describes the agents architecture, and Sec. 4 describes agents actions in more detail. The paper concludes with a discussion of a prototype implementation and with a comparison of related work on conversation analysis and computational dialectic.

2 Global Setup

The global setup consists of a set $\mathcal{A} = \{A_1, \dots, A_n\}$ of agents ($n \geq 2$) and a public communication medium T, called *the table*. T can be seen as a blackboard, or as "open air," by means of which agents are able to exchange messages in public. More specifically, T is a passive object with two essential methods, viz.

```
put(m: message)
get(t1: time, t2: time): setofMessages
```

By way of the second method agents can retrieve all messages that were uttered between time points t_1 and t_2.

Experiments are performed in runs. A run is a complete session in which agents are initialized by the programmer, and then exchange messages autonomously until no agent activity is observed within some fixed time period. At the start of each run each agent receives a number of propositions from the programmer to fill its belief base with. The initial goal base of each agent cannot be programmed and consists of one action, viz. `listen`. A typical run starts with one or more agents that have a computed interest in determining the credibility of one or more propositions. These propositions are put on the table in the form of queries. These queries invoke a dispute. This dispute ends as soon as all agents have lost all incentives to utter speech acts (typically queries and answers to queries).

For the sake of simplicity the present setup assumes that agents comply to specific (and admittedly often unrealistic) maxims of co-operation. In particular, it is assumed that agents are honest and credulous. Honesty corresponds to the

Gricean maxim that agents are forbidden to put forward information they do not believe; credulousness corresponds to the property that agents believe what they are told. I do not think that it is difficult to extend the present setup to a scenario where these constraints are dropped. Evidently this avenue goes beyond the scope of this paper.

For the same reasons of simplicity, the model does not assume that agents are perfect reasoners or communicators. In particular an agent can be programmed such that it prioritizes communication at the expense of logical inference. Conversely, it is possible to program "ponderers" that prioritize internal inference at the cost of communication. Obviously, both extremes are undesirable and the programmer is responsible for achieving the right balance. The programmer can achieve this balance by ensuring that (most) communication actions invoke logical actions and conversely (which is a natural phenomenon).

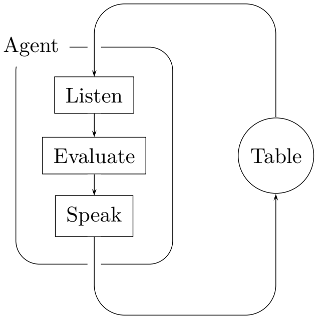


Fig. 1. Serial deliberation

3 Agent Architecture

An agent $A = (B, P, G)$ is a daemon that possesses a declarative belief base B, a procedural belief base P, and a goal base, or agenda, G.

- The declarative belief base B contains propositions about the state of the world, formulated in a simple logical object language, annotated with information that pertains to the proposition's origin, the proposition's degree of belief, and other attributes.
- The procedural belief base P contains information that is concerned with internal procedural matters. An example of a procedural item is a (private) method that returns a pointer to the next unread message (a so-called *bookmark*).
- The goal base G is a (private) priority queue filled with actions. Actions can either be internally or externally directed. Thus, agents can schedule belief updates as well as sending messages.

Agents are not able to inspect, modify, or communicate about the contents of P. Therefore, the objective in designing A is to put as much as possible of P in B and G, so that agents can reason and communicate about their beliefs.

3.1 Deliberation Cycle

Each agents runs an eternal loop, also called a *cycle* [3], or a *deliberation cycle* [6, 12]. Contrary to first-generation deliberation cycles this loop is not serial

(Fig. 1) but prioritized (Fig. 2).[2] At every pass of the loop the agent takes an action from the priority queue and executes it. A typical execution of an action amounts to doing a few operations in the internal representation format of the agent, interspersed with (or followed by) scheduling some new actions. The priority of these new actions depends on their type and the priority and contents of the action that caused the scheduling of the new actions.

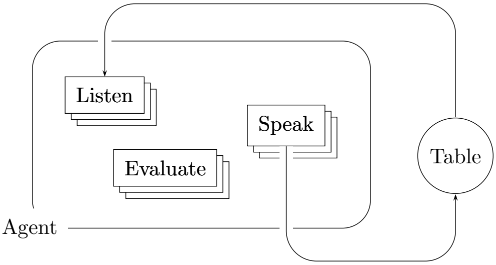


Fig. 2. Prioritized deliberation

Actions are not executed in the order in which they are put on the agenda, but according to they priority. If a set of actions nevertheless must be executed in succession, this can be accomplished in two ways.
(1) The first way is to simply concatenate the actions as a plan in the body of the action statement a. In this way, all actions in the body of a are executed immediately if the head of a is taken from the priority queue.
(2) The second way is to assign decreasing priorities but equal activation factors to a list of actions. In this way the actions are guaranteed to be executed in succession, be it that they are likely to be interleaved with other actions. For many types of actions this is no problem.

An example of a typical action is "Listen" (Action 1). When this action is taken from the queue, the agent fetches the last unread message from the table. If this action succeeds, and the message is not from the agent itself and not addressed to another agent, the agent schedules a "process-message" activity. Independently a next listen activity is scheduled.

3.2 Goal Base

An agent's goal base, or *agenda*, is a priority queue filled with actions. To ensure that all actions are eventually executed, this priority queue is equipped with a scheduler that is derived from the standard priority schedulers used in operating systems theory [27].

The scheduler works as follows. Contrary to [3, 6, 12], actions do not have preconditions but possess, besides an action-inherent priority, a so-called *activation factor*. (If one of them is missing, a reasonable default is used.) When the agenda is initialized, the agenda is given its own activation factor as well. The activation factor of an agenda represents the nominal speed with which scheduled actions rise ("bubble") to the top of the priority queue, once they are put on the agenda. Thus, agendas as well as prioritized actions possess an activation factor.

Each time an agent puts a new action a on its agenda, the priority of a is increased with a's activation factor times the activation factor of the agenda. The

[2] Cf. relation work in Section 7.

Action 1. Listen

```
table.get( procedural.first-unread-message )
if defined message then
    if message.from == my-own-name then
        Purge message # because it is my own
    else if defined message.to $\wedge$ message.to != my-own-name then
        Purge message # not addressed to me
    else
        $G$.schedule( “process-message”, message )
else
    Pass
$G$.schedule( “listen”, :priority⇒-5, :activation⇒1 )
```

result of this mechanism is that actions with a low priority and a high activation factor will rise relatively fast to the top of the agenda during successive insertions. Conversely, actions with a low activation factor will probably remain on the agenda for a long time, unless they were already given a high priority from the start. Actions that should receive low priorities but high activation factors are typically low-level actions that must be executed on a regular basis “to keep an agent going,” without blocking the more important high-level actions. Listening is an example of such an action (and the only example in my model). Conversely, high-level actions, such as logical inference and inquiry, typically receive a high priority but a low activation factor.

3.3 Belief Base

Each agent possesses a private belief base B that is only filled with propositions. A proposition is an object with a number of attributes as described in Table 1.

Propositions can take the form of an atom, a literal, a rule, or the negation of a rule. (The latter is established by naming rules, and then negating the name of the rule.) In the current model the degree-of-belief of a proposition is an element in $[0, 1]$ that indicates to what extent an individual agent believes in that proposition. The degree-of-support of a proposition is an element in $[0, 1]$ that indicates to what extent a proposition receives logical support from other propositions via logical inference. The rules for propagation of support through rules of inference are primitive but provide sufficient material to construct an elementary logic for agents.

A trivial example of a proposition is the atomic proposition P. This proposition has the following slots filled: name, DOB, DOS, supports, supported-by, claimants, questioners, and last-questioned-by. The following slots are empty (and stay empty): antecedent, consequent, strength and RDOS. The rest of the slots are filled optionally.

A non-trivial example of a proposition is the negation of the rule

$$r : P, Q \rightarrow S.$$

The slots negation, DOB, DOS, supports, supported-by, claimants, questioners, and last-questioned-by are filled. The negation slot points to the proposition object with name r. The following slots are empty (and stay empty): name, antecedent, consequent, strength and RDOS. The rest of the slots are filled optionally.

Table 1. Proposition

Key	*Description*	*Accessibility*
name	agent's private name for this proposition	optional
negates	reference to proposition that is negated	optional
importance	number that indicates how much importance the agent attaches to the credibility of this proposition	default 1
dob	degree of belief $\in [0, 1]$	default 0.0
DOS	degree of support $\in [0, 1]$	default 0.0
supports	list of references to internal propositions supported by this proposition	default []
supported-by	list of references to internal that support this proposition	default []
claimants	list of agent names that have claimed this proposition	default []
questioners	list of agent names that have questioned this proposition	optional
last-questioned-by	agent that questioned this proposition last	optional
consequent	head of rule	optional
antecedent	body of rule	optional
strength	rule strength	default 1.0
RDOS	degree of support running through this rule	default 0.0

The relation between P and r is that P occurs in the antecedent list of r while r occurs in the supported-by list of P. In the line of Toulmin [29] rules can support or deny other rules. Thus, the consequent of a rule can be another rule, or the negation of another rule (called *undercutter* in Pollock [20] and subsequent work in computational dialectic). (Explained in more detail in overview articles such as [5, 32].)

3.4 The Underlying Argumentation Model

Internally, agents try to enhance their support of selected propositions by means of arguments.

The underlying argumentation model that I use for the larger MAS is a trimmed down version of formalisms as proposed in, e.g., [1, 13, 22, 33]. According to these formalisms, arguments are obtained by chaining rules into trees, and arguments supply different degrees of support to their conclusions. How support is computed depends on the modalities of the various rules and propositions, and how we think these modalities should propagate through an argument. In

Table 2. Message

Key	*Description*	*Accessibility*
id	message id (assigned by table)	optional
from	sender	optional
to	addressee	optional
content type	query | statement	optional
subject	the proposition that the message is about	optional
priority	priority, as perceived by the sender	optional
consequent	if the message is a justification, this field will store the consequent of that justification	optional
antecedent	if the message is a justification, this field indicates its antecedent	optional
strength	if the message is a justification, this field indicates the strength as perceived by the sender	optional
dob	degree of belief as perceived by the sender	optional
DOS	degree of support as perceived by the sender	optional
reference	message to which the present message refers to	optional

reality these modalities are often qualitatively specified ("weakly," "strongly," ..., "certainly") or even plain absent. For the sake of simplicity, my model assumes that modalities are elements of the real interval $[0, 1]$ and that modalities are always present on places where we expect them to be specified.

Definition 1 (Support). *Let σ be an argument.*

1. *If σ is a singleton argument, i.e., if σ is of the form $\sigma = \{p\}$ where p is a proposition, then the* degree of support *of σ is equal to the degree of belief of p:*
$$DOS(\sigma) =_{Def} DOB(p)$$
2. *If σ is a compound argument with conclusion p, top-rule $r : p \leftarrow\!\!(s)\!\!- p_1, \ldots p_n$ and sub-arguments $\sigma_1, \ldots, \sigma_n$, then the* degree of support *of σ is given by*

$$DOS(\sigma) =_{Def} \max\{\ DOB(p) \\ \min\{\ DOB(r), s * \min\{\ DOS(\sigma_1), \ldots, DOS(\sigma_n)\}\}\ \} \tag{1}$$

The rationale behind (1) is the so-called *weakest link principle*, which says that every construction (in this case: every argument) is as strong as its weakest link. A convincing justification for the weakest-link principle can be found in work of Pollock [20, 21]. Another principle that I have followed is that rules propagate support with an amount that is proportional to their strength. I immediately admit that (1) is an overly simplistic account of support. Nevertheless, the reason to use (1) is that it provides agents with just enough logical machinery to perform simple defeasible reasoning internally, and to engage in simple dialogues about their own defeasible knowledge externally.

Example 1 (Propagation of support). Consider the following set of propositions.

prop	DOB	*prop*	DOB	*prop*	DOB	*name*	*rule*	DOB
a		b	0.7	c		r_1:	$a \leftarrow\!(0.5)\!- b, c$	1.0
d	0.2	e		f	0.8	r_2:	$b \leftarrow\!(0.5)\!- d, e$	1.0
g	0.1	h	1.0	i	0.6	r_3:	$c \leftarrow\!(0.5)\!- f, g$	1.0
j	1.0					r_4:	$e \leftarrow\!(0.5)\!- h, i$	1.0
k	0.8					r_5:	$g \leftarrow\!(0.5)\!- j, k$	1.0

Thus, we have eleven atomic propositions and five propositions of type rule, or simply: rules. If all reasons are chained together, we obtain a representation of an argument as displayed in Fig. 3. Agents spend resources in trying to discover such arguments through backward chaining and to propagate support forwards (upwards in Fig. 3) in case they receive new information about the credibility of a specific proposition.

How agents schedule inference and communication actions that relate to support is further explained in Sec. 4.2.

4 Actions in More Detail

In the current implementation agents can schedule and execute approximately forty-five different actions, ranging from actions that are concerned with internal inference to actions that are concerned with communication. This section discusses the interaction between the different types of actions and explains how this interaction shapes the discussion. For reasons of space, I do not review all actions.

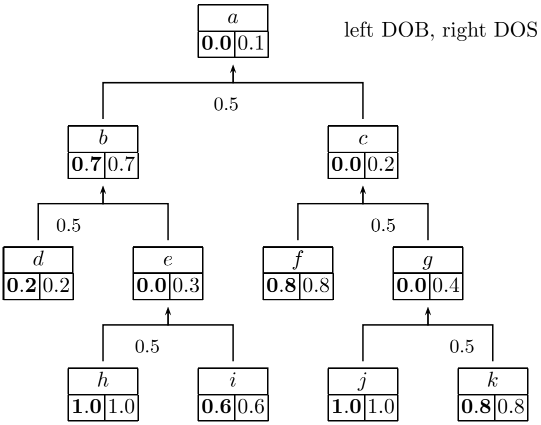


Fig. 3. Propagation of support

Roughly there are three categories of actions. The first category of actions is epistemic and is concerned with inquiry prioritization (which propositions to investigate next), logical inference, and belief updates. Examples of such actions are "propagate-degree-of-support-for p" or "compute-degree-of-support-for p," where p is a proposition. Other actions relate to the external world and are concerned with speaking and listening. A third category of actions is concerned with linking the external to the internal world. Examples of these type of actions are actions to process or incorporate messages and translate them to proposition objects.

An important property of the model is that agents decide autonomously which issues need to be discussed, when to query other agents about issues, when to keep on querying other agents, and when to settle for an answer.

4.1 Inquiry

New queries arise due to a combination of importance and epistemic dissonance [4,24]:

$$\textit{urgency-to-enquire}(p) = \textit{importance}(p) * \mathrm{DOS}(p) * \mathrm{DOS}(\neg p) \tag{2}$$

In general, epistemic dissonance is the degree of conflict between two or more competing beliefs, of which at least one belief is deemed important for reasons that may be external to the logical or epistemological formalism (for example for practical reasons) [9,19]. Here, the epistemic dissonance of a proposition p is simplified into a simple mathematical product.

In the present model, importance is an external factor that, if absent, defaults to 1.0. The principle of epistemic dissonance can be used as a threshold to decide whether it is allowed to query others if there is doubt concerning a proposition that cannot be resolved on the basis of an agent's private beliefs. For example, if the threshold is set to 0.8 then propositions are queried once *urgency-to-enquire*(p) > 0.8. This leads to an elementary Action 2.

Action 2. Inquire(p: proposition, i: priority)

```
$G$.schedule( “compute-degree-of-support-for”,
  $p$ :priority⇒ $i+1$, activation⇒1 )
$G$.schedule( “compute-degree-of-support-for”,
  $\neg p$ :priority⇒ $i+1$, activation⇒1 )
if $\textit{urgency-to-enquire}(p) \geq 0.8$ then
    $G$.schedule( “query”, $p$ :priority⇒ $i$, activation⇒1 )
```

The priority settings in Action 2 enforce that the urgency-to-enquire is computed only after the agent did an internal search into its own beliefs on the credibility of that proposition.

4.2 Inference

Inference amounts to all actions that are internal to an agent and are aimed to enhance the degree of support of propositions.

It must explicitly reiterated here that the underlying agent object logic is extremely simplistic and only serves as a vehicle to demonstrate what agents can do (and are supposed to do) if they engage in a discussion. More mature theories of belief revision are to be found in philosophical logic [10] and the theory of Bayesian belief updates [18].

Basically, there are two categories of inference actions, namely *pull* and *propagation* (or: pull and push).

Propagation is best explained in terms of belief updates related to incoming messages. If an incoming message on a proposition p reports on a higher degree of belief in p, then the receiving agent schedules an update to the degree of support of its internal representation of p (remember that agents are credulous). This update amounts to serially propagating the new degree of belief via (internal) rules to other (internal) propositions. The actual propagation is scheduled as well, so that propositional belief updates are interleaved with other actions. This all depends on the priorities that are attached to the belief update actions. Thus, it may well happen that an agents accidently reports misinformation because it has it given a low priority to its internal belief update actions. (This behavior can occasionally be enforced in the implementation by setting the start priority of belief updates to a low values.)

Belief pull corresponds to the informal question "what do I actually know about p?" and is the result of an inquiry action (Action 2). Belief pull comparable to backward chaining, with the restriction that agents must at each cycle decide whether to search further backwards for justification, or to execute other actions first. The present model solves this by attaching a priority to every backward chaining action that is a function of the priority of the original action and the expected maximal return of support.

4.3 Query

A query is a request for information about a particular proposition. Queries can be open or addressed to a particular agent.

Open queries have no explicit addressee and can be taken up by any agent that finds it important enough to process it. When an agent decides to query other agents (compare Eq. 2), it composes a message with the name of the proposition and a token indicating that the message emitted is a query.

An addressed, or directed, query is a request to a specific agent to explain or justify a certain claim. The present model works with open queries only. On the basis of the message format and the deliberation cycle mechanism it is safe to predict that the existing model can be naturally extended to an agent model in which agents know how to deal with addressed messages.

4.4 Response

Two types of messages can appear on the table, viz. queries and statements. Queries have been discussed above.

A statement is simply a public announcement of an agent in which it declares that it believes in a certain proposition to a certain degree of belief. Analogous to queries, claims can be addressed to a particular agent, typically as an answer to a previous query. Alternatively, claims can be addressed to no agent in particular. Such open claims can be seen as theses, or positions, meant to lure other agents into a discussion. The present model works with directed statements only. Further, the present model allows agents to update their beliefs with statements (answers) that are directed to other agents. This possibility to overhear messages that are aimed at other agents and to respond to such messages is a more or less arbitrary commitment of the architecture.

The three essential actions in forming replies are Action 3, 4 and 5.

Action 3. Process-query(m: message)

```
B.incorporate-query( $m$ )
reply = fabricate-reply( $m$ )
**if defined** reply **then**
    $G$.schedule( “speak”, reply )
    $G$.schedule( “process-query”, $m$ )
**else**
    $d$ = Message.new( :subject→no-answers, :referent→ $m$ )
    $G$.schedule( “speak”, $d$ )
```

Action 4. Fabricate-reply(m: message)

```
reply = next-unpublished-answer-to( $m$.subject )
**if defined** reply **then**
    **return** reply.into-message-format
**else**
    **return** nil
```

Action 5. Next-unpublished-answer-to(s: subject)

```
prop = $B$.prop-retrieve( $s$ )
**if defined** prop **then**
    **return** prop **unless** $P$.published(prop)
**for** rule $\in$ $B$.rule-retrieve( $s$ ) **do**
    **return** rule **unless** $P$.published(rule)
prop = $B$.prop-retrieve( $s$.negation )
**if defined** prop **then**
    **return** prop **unless** $P$.published(prop)
**for** rule $\in$ $B$.rule-retrieve( $s$.negation ) **do**
    **return** rule **unless** $P$.published(rule)
```

With Action 3, the query is incorporated in B first. This means that the agent creates a corresponding proposition in B (if such a proposition does not exist yet), stores the name of the agent that queried the proposition and the time t that this particular proposition is queried. If there is an unpublished answer, then the receiving agent schedules a speech act in which it emits an answer and schedules a new action to process this query (for there may be more answers). If there are no more answers left, only a speech act is scheduled in which the agent effectively says that it has no answers, either because it has no answers to begin with, or else because it ran out of answers.

With Action 5, the action P.published/1 is a check on the procedural belief base in which the agent verifies whether an agent (including the agent itself) already has published the proposition in question.

5 Implementation

To allow experiments with different set-ups, and to see whether the generated dialogues make any sense, I have implemented the model in the oo-scripting language Ruby. The purpose of the implementation is to experiment with different inputs and with different parameter settings.

The results experiments can be reproduced with the help of an online prototype of which the URL is given at the end of this section.

5.1 Experiments

This section presents a simple example in which a group argues about the credibility of a certain proposition. The example is simple in that it does not involve negation and auto-inquiry has been turned off for the sake of brevity and readability. Examples with slightly more complex input already stir up an emormous amount of actions and messages, so that the structure of the dialogue becomes lost in the output. The reader is invited to try out more complex input at the URL mentioned above.

Suppose we have three agents, Alice, Bob, and Charles, and suppose that Bob is instructed to issue a query on C (Fig. 4).

Fig. 5 shows a trace of the run. First Bob tries to find out how much C is supported by its own beliefs. Then it decides to ask others about C. This query is not related to previous messages, hence the empty reference `--`. The two other agents process this question and burrow into their own beliefs to discover to what extent they support C themselves. Alice responds with a justification. This justification is received by Bob. Since the justification end in B and Bob has no support for B, Bob decides to query further and ask others about B (line three). Charles explains B with $A \rightarrow B$. Finally, Bob says "ok' at line five because it can connect the antecedent of $A\ 0.7 \rightarrow B$ to its own support for A.

For reasons of space. the results displayed here are rather minimal. The reader is therefore invited to experiment online at `http://www.cs.uu.nl/~gv/code/liberal`. The online prototype is supplied with a Java-doc style documentation and the code itself can be downloaded if desired.

```
Agent Alice
B 0.8=> C

Agent Bob
C?
A 0.9

Agent Charles
A 0.7=> B
```

Fig. 4. Input

```
   Bob thinking, enquire
   Bob thinking, compute_dos_for
   Bob thinking, speak
1. Bob [--]: Why C?
   Charles thinking, handle_question
   Alice thinking, handle_question
   Alice thinking, speak
2. Alice [1]: C, since B
   Bob thinking, incorporate
   Alice thinking, interpret
   Bob thinking, compute_rdos_for
   Alice thinking, handle_question
   Bob thinking, propagate_rdos_of_rule
   Bob thinking, question_antecedent
   Bob thinking, attack_antecedent
   Bob thinking, question_antecedent_element
   Bob thinking, speak
3. Bob [2]: Why B?
   Charles thinking, handle_question
   Alice thinking, handle_question
   Charles thinking, speak
4. Charles [3]: B, since A
   Bob thinking, incorporate
   Charles thinking, interpret
   Bob thinking, compute_rdos_for
   Charles thinking, handle_question
   Bob thinking, propagate_rdos_of_rule
   Bob thinking, question_antecedent
   Bob thinking, attack_antecedent
   Bob thinking, question_antecedent_element
   Bob thinking, speak
5. Bob [4]: Ok
```

Fig. 5. Summary of run

6 Results

During the experiments, I noticed that all discussions terminate. This can be understood as follows. Firstly, a finite number of queries may be linked to a finite number of answers. Further, agents keep an account of which queries they have answered. Since, queries are dealt with at most once, termination is ensured for ach individual agent. Since a MAS contains a pre-determined of agents by definition, eventually termination is ensured for the entire MAS.

I also observed that agents will reach a conclusion on accessible facts within a reasonable amount of turns. This can be explained by the fact that explanations (i.e., explanatory rules) cannot be chained indefinitely. A a consequence each justification has a stopping place, so that agents will either accept facts or abandon search on explained statements within a bounded number of dialogue moves.

Properties such as termination and response are proven formally in [17]. Intuitive results reported there indeed correspond with my model albeit my judgement is based on observation rather than on model analysis. Other results do not correspond to my model, for example that credulous agents can be convinced of everything, even of propositions contrary to their beliefs [17, Prop. 6.8, p. 367].

Even though discussions terminate, I noticed that traces of runs are extremely long, even for trivial input. This observation points to two further research problems.

1. The problem to maintain overview on the activity in a MAS.
2. Estimating the number of actions in a MAS based on the size of the input.

Investigation of these problems falls beyond the scope of this paper, but is briefly discussed in Section 8.

7 Related Work

The term of liberal dispute was earlier coined by Prakken in an article on relating protocols for dynamic dispute with logics for defeasible argumentation [23]. In Prakken's work, a liberal dispute is an exchange of arguments (rather than an exchange of propositions as is done in this paper) such that every move is relevant (in Prakken's sense) to the first argument in that dispute. The main effort in Prakken's work is to prove that liberal protocols are sound and fair. It is possible to prove such a result because the formalism assumes that arguments are exchanged in their entirety and that participants in a discussion eventually respond to all utterances that are logically connected to their beliefs. In turn, these assumptions rest on the hypothesis that agents are logically omniscient, communicate everything they know and are able to process everything they receive. The model presented in this paper is less idealistic and thus cannot guarantee such a result.

Although it is arguably one of the simpler types of dialogue, inquiry has received less attention than negotiation or persuasion. An exception is the work by McBurney and Parsons [14] on scientific investigation. Our purpose is very similar to theirs. They describe a *Risk Agora*, as they call it, that allows the storage of multiple arguments for and against some claim. However, they do not treat multi-party issues explicitly. The Agora is an asynchronous channel; no coordination rules are given.

My present work also relates to the Newscast protocol [30]. The Newscast protocol is a kind of 'gossiping' protocol that can be used to disseminate information in distributed systems. A difference is that the newscast protocol can only pass on information. No mechanism exists to specify queries. The Newscast protocol is also implemented and experimented with albeit on a much larger scale, and the results are reported quantitatively.

Recently, researchers in the European SOCS project proposed a model of agency for global computing called the KGP model (knowledge, goals and plans) [3, 26]. This model is particularly interesting because a number of researchers that worked on this model have a strong background in argumentation. The KGP model proposes a logical architecture that is concerned with agents that (for various reasons) have incomplete information about their environment, and want to update that information by engaging in a conversation with other agents. Like the model that is proposed in this paper, KGP uses uses priorities by defining preference policies over the order of application of transitions. However, the prioritization is more complex because entire logic programs are prioritized rather than atomic actions. Every KGP-agent contains an argumentation component that is a direct derivative of the classical argumentation theories that have preferred and admissible sets as their semantics. It is remarkable that, in other publications, some of these authors argue that finding admissible and preferred arguments can be very hard [7].

The lightweight version of 3APL, called 3APL-M does have a so-called plan ranker [6]. This an internal class, part of the planner sub-system, which classifies the plans in the plan base by calculating its utilities. This component drops the plans that have negative utility from the Plan Base.

8 Future Work

A problem that I noted with ourthe experiments is that it is difficult to monitor all the action. At present all activities are written to a linear log but this solution is unsatisfactory from multiple viewpoints, even for small input. Although there exist tools to monitor agent communication (e.g., JADE's message sniffers [16]), a larger problem is to monitor all pre-processing prior to message emission and all processing of messages once they are received. Currently, I have colored the output to create a global distinction. Each agent possesses its own color. Dark colored log entries relate to internal processing, while light colored log entries relate to agent activity that are more related to communication. Currently there are four such color categories.

9 Conclusion

In this paper I proposed a liberalized version of existing argumentation protocols. Within the resulting setup agents can construct arguments autonomously by participating in an inquiry dialog, thus bringing ideas of computational dialectic to bear in a multi-party inquiry. It is the connection between the two disciplines that counts here. Obviously more work has to be done to consolidate and utilize this connection.

Acknowledgement. I'd like to thank Martin Caminada, Mehdi Dastani, Henry Prakken and two anonymous referees for their helpful comments. This research was supported in part by a European Commission STReP grant ASPIC IST-FP6-002307. This project aims to develop re-usable software components for argumentation-based interactions between autonomous agents.

References

1. P. Baroni, M. Giacomin, and G. Guida. Extending abstract argumentation systems theory. *Artificial Intelligence*, 120(2):251–270, 2000.
2. Robbert-Jan Beun. On the generation of coherent dialogue: A computational approach. *Pragmatics & Cognition*, 9(1):37–68, 2001.
3. Andrea Bracciali, Neophytos Demetriou, Ulle Endriss, Antonis Kakas, Wenjin Lu, Paolo Mancarella, Fariba Sadri, Kostas Stathis, Giacomo Terreni, and Francesca Toni. The KGP model of agency for global computing: Computational model and prototype implementation. In *Proc. of the Global Computing 2004 Workshop*, volume 3267 of *LNCS*, pages 342–369. Springer Verlag, 2004.
4. Urszula Chajewska and Joseph Y. Halpern. Defining explanation in probabilistic systems. In *Proc. of the 13th Conf. on Uncertainty in Artificial Intelligence*, pages 62–71, 1997.
5. Carlos I. Chesñevar, Ana G. Maguitman, and Ronald P. Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
6. Mehdi Dastani, Frank de Boer, Frank Dignum, and John-Jules Meyer. Programming agent deliberation: An approach illustrated using the 3APL language. In *Proc. of the Second Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'03)*, 2003.
7. Yannis Dimopoulos, Bernhard Nebel, and Francesca Toni. Finding admissible and preferred arguments can be very hard. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 53–61. Morgan Kaufmann, 2000.
8. Sylvie Doutre and Jérôme Mengin. On sceptical vs. credulous acceptance for abstract argument systems. In *Tenth International Workshop on Non-Monotonic Reasoning (NMR 2004)*, pages 134–139, 2004.
9. N. Everitt and A. Fisher. *Modern Epistemology: A New Introduction.* McGraw-Hill, 1995.
10. Peter Gärdenfors. *Knowledge in Flux: Modelling the dynamics of epistemic states.* MIT Press, London, 1988.
11. Charles L. Hamblin. Mathematical models of dialogue. *Theoria*, 37:130–155, 1971.
12. Koen V. Hindriks, Frank S. de Boer, Wiebe van der Hoek, and John-Jules Ch. Meyer. Agent programming in 3apl. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999.
13. Fangzhen Lin and Yoav Shoham. Argument systems: A uniform basis for nonmonotonic reasoning. In R.J. Brachman, H.J. Levesque, and R. Reiter, editors, *Proc. of the 1st Int. Conf. on Knowledge Representation and Reasoning*, pages 245–255. Morgan Kaufmann Publishers, 1989.
14. Peter McBurney and Simon Parsons. Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and Artificial Intelligence*, 32(1):125–169, 2001.
15. Peter McBurney, Simon Parsons, and Michael Wooldridge. Desiderata for agent argumentation protocols. In *Proc. of the First Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 402–409. ACM Press, 2002.
16. Pavlos Moraitis and Nikolaos I. Spanoudakis. Combining gaia and jade for multiagent systems development. In *Proc. of the 17th European Meeting on Cybernetics and Systems Research (EMCSR 2004)*, April 2004.
17. Simon Parsons, Michael Wooldridge, and Leila Amgoud. Properties and complexity of some formal inter-agent dialogues. *The Journal of Logic and Computation*, 13(3):347–376, 2003.

18. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, Inc., Palo Alto CA, 2 edition, 1994.
19. J.L. Pollock. *Knowledge and Justification.* Princeton University Press, 1974.
20. John L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person.* MIT Press, Cambridge, MA, 1995.
21. John L. Pollock. Implementing defeasible reasoning. Presented at the Computational Dialectics Workshop, at FAPR'96, June 3-7, 1996, Bonn. Cf. http://nathan.gmd.de/ projects/zeno/fapr/programme.html., 1996.
22. H. Prakken and Gerard A.W. Vreeswijk. Logics for defeasible argumentation. In D.M. Gabbay et al., editors, *Handbook of Philosophical Logic*, pages 219–318. Kluwer Academic Publishers, Dordrecht, 2002.
23. Henry Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. In Shahid Rahman and Helge Rückert, editors, *New Perspectives in Dialogical Logics*, volume 127, pages 187–219. Synthese, 2001.
24. Anand S. Rao. Integrated agent architecture: Execution and recognition of mental-states. In *Intelligent Agent Systems: Theoretical and Practical Issues*, volume 1087 of *Lecture notes in computer science*, pages 159–173. Springer-Verlag, Berlin, 1996.
25. John R. Searle. Conversation. In J.R. et al. Searle, editor, *(On) Searle on Conversation*, pages 7–30. John Benjamins, 1992.
26. Kostas Stathis, Antonis Kakas, Wenjin Lu, Neophytos Demetriou, Ulle Endriss, and Andrea Bracciali. Prosocs: A platform for programming software agents in computational logic. In J. Müller and P. Petta, editors, *Proc. of the 4th Int. Symposium "From Agent Theory to Agent Implementation" (AT2AI-2004)*, April 2004.
27. Andrew S. Tanenbaum. *Operating Systems: Design and Implementation (Second Edition).* Prentice Hall, 1997.
28. Jasper A. Taylor, Jean Carletta, and Chris Mellish. Requirements for belief models in co-operative dialogue. *User Modelling and User-Adapted Interaction*, 6:23–68, 1996.
29. Stephen Toulmin. *The Uses of Argument.* Cambridge University Press, 1985.
30. Spyros Voulgaris, Márk Jelasity, and Maarten van Steen. A robust and scalable peer-to-peer gossiping protocol. In *Proc. 2nd Int. Workshop on Agents and Peer-to-Peer Computing (AP2PC 2003)*, 2003.
31. Gerard Vreeswijk and Joris Hulstijn. A free-format dialogue protocol for multi-party inquiry. In Jonathan Ginzburg and Enric Vallduví, editors, *Proc. of the Eighth Int. Workshop on the Semantics and Pragmatics of Dialogue (Catalog '04)*, pages 273–279, 2004.
32. Gerard Vreeswijk and Henry Prakken. Credulous and sceptical argument games for preferred semantics. In Ojeda-Aciego *et al.*, editor, *Proc. of the 7th European Workshop on Logics in Artificial Inteligence (JELIA 2000)*, volume 1919 of *LNCS*, pages 239–253. Springer-Verlag, 2000.
33. Gerard A.W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90:225–279, 1997.
34. Gerard A.W. Vreeswijk. Liberalizing protocols for argumentation in multi-agent systems. In *Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1259–1260, New York, NY, USA, 2005. ACM Press.

Protocol Synthesis with Dialogue Structure Theory

Jarred McGinnis, David Robertson, and Chris Walton

Centre for Intelligent Systems and Applications
University of Edinburgh
11 Crichton Street
Edinburgh, Scotland EH8 9LE
j.p.mcginnis@sms.ed.ac.uk

Abstract. Inspired by computational linguistic approaches to annotate the structures that occur in human dialogue, this paper describes a technique which encodes these structures as transformations applied to a protocol language. Agents can have a controlled and verifiable mechanism to synthesise and communicate their interaction protocol during their participation in a multiagent system. This is in contrast to the approaches where agents must subscribe to a fixed protocol and relinquish control over an interaction that may not satisfy the agent's dialogical needs or rely on internal its reasoning to determine which message to communicate at a certain point in the dialogue.

1 Introduction

Research into agent communication is producing increasingly more robust models. Much of this research has turned to other disciplines for inspiration. Philosophy and Linguistics have a several thousand year head start in reflecting upon the nature of communication [1]. These thinkers are concerned with human communication in particular, but insights and models they have developed are readily applicable to the study of agent communication. BDI-logics [2],speech acts [3], social commitment [4] and argumentation [5] have originated in the works of philosophers and linguists [6, 7, 8].

Many have been attracted to a societal view of communication. They take the position that communicating entities, whether they be organic or synthesised, do not communicate in a vacuum but rather in the context of the society made of the other communicating entities around it. This society has rules which govern the behaviour of the agents, constraining the members to perform in accordance with a set of implicit or explicit protocols. Participants in the society willingly sacrifice autonomy and submit to these protocols in order to gain a measure of utility or to accomplish a goal of more value than the independence lost. Traditionally, protocols have been seen as static and inflexible and defined as specifications for a human engineer to interpret and encode his or her agent with that interpretation. The approach described in this paper addresses the possibility of a protocolled approach to communication where the agents themselves not only communicate the protocol to each other but also create the protocol during the interaction.

Protocols are not only created with respect to societal conventions but the act of communication itself has conventions to which speakers adhere. Linguists have been

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 199–216, 2006.

interested in this phenomena.They have also adopted this study for the purposes of annotating human dialogue for the purposes of automated text analysis and generation [9]. The challenges this field faces largely differ from the concerns of multiagent communication, such as anaphoric ambiguities, but there are findings which add a robustness to the protocolled societal approach to agent communication.

Relationships exist between messages regardless of the particular domain with which the messages are concerned. A question implies the anticipation of the eventual occurrence of an answer even if the reply is a shrug of the shoulders. This is regardless of whether that answer be to the question of ”What time is it?” or ”Can you compare and contrast the post-modern interpretations of abstract expressionism to a random sequence of adjectives?” It is these generalised patterns which exist in human communication that we have adopted for our purposes. The result is the creation of a means to synthesise a protocol which can reproduce the reliable communication of other protocolled approaches without being fixed to a static protocol.

We will begin our discussion with a introduction and explanation of dialogue structures in general and within the context of multiagent communication. Section 3 explains the essentials to the protocol language used to implement this approach. The library of transformations which enable the synthesis of protocols is described in section 4. The process of synthesising protocols is introduced and illustrated in sections 5and 6. Finally, we conclude in section 7.

2 Using Dialogue Structures

In human dialogue the utterances that the participants make do not occur in isolation. Humans rely on tacit patterns to ground communication. Some have proposed this is the following of certain rules, and others have argued these rules are only descriptions of the process of having a conversation [7]. Regardless,these patterns can be generalised without concern to the content of the messages. The idea for this approach was largely inspired by the works of [9, 10, 11], and the standardisation efforts of Dialogue Structure Theory (DST) for the annotation of human dialogue transcriptions.

There are a number of approaches that more or less could be used for the run time synthesis of interaction protocols. Although each have proved their worth for a variety of multiagent applications. Each fails in some aspect to provide the unique advantages found by the use of dialogue structures.

Performatives are a common approach for agent communication, and it may be possible to pack pan-dialogical concerns into individual performatives. Yet, this would be an ungainly implementation and an abuse of the spirit of performatives. They are meant to reflect the conditions and effects of a single communicative act rather than the relationships between them or their place within a sequence of message exchanges. Our concern is more generic than particular performatives in a given ACL. It is our goal to capture the generic structure of conversation that occurs in dialogues regardless of the performative or ACL used.

Planning research has been brought to bear on the problem [12]. Agents use planning techniques to produce an interaction protocol to reach a previously defined goal by means of communicating with other agents. Firstly, communication is not always driven by clearly defined *a priori* goals. It could be the goal of the dialogue to determine the goal of the interaction.Planning is also presented with the unique challenges of the agency model. Besides planning's reputation for a paucity in terms of data structures, there is also another difficulty in using planners for this purpose. It will be difficult for a planner to produce anything more robust than a look-ahead planner, because of the unpredictability of other agents. The planning agent would constantly be replanning in reaction to others' actions. It would result in a lot of computation without much satisfaction. Even with the help of making assumptions about other agent's rational behaviour, existing approaches still have speed issues for real-time systems. It is for this reason that it would be much more appropriate to have a small set of transformations which the agent can apply mechanically to achieve the same goal. This is exactly what we describe in this paper.

Machine learning is also being applied to the agency paradigm [13]. The techniques of machine learning introduce a number of unnecessary difficulties. For example, it would be helpful to have transparency and readability of the protocols used by the agents to facilitate human/computer interaction or simply to enable humans to understand the protocols used which will assist in the design of new agents. Also, the common problem of producing corpora that hounds machine learning for agency is also a problem in synthesising interaction protocols. Similar to planning approaches,the same goal can be achieved with a set of transformations which can free the agent to spend its computation on learning a strategy for the domain rather than the discussion of that domain.

It is correct to point out the work using social commitments, norms,dialogue games, and other such models of communication provides agents with the ability to reason about communication. It is not the goal to replace any particular model of agency. The goal is to exploit the unique advantages provided by the LCC language and framework, but to enhance its flexibility. The transformations are purely dialogical in the sense they are generic operations which unfold a single message protocol to a two message protocol which in turn can be used to synthesise a three message protocol, and so on. The agent receiving the synthesised protocol can follow it blindly without needing to understand that its dialogical actions satisfy some commitment, norm,or rule of a dialogue game. This is the advantage being touted. A clean and simple dialogically driven means to drive protocolled communication while maintaining an agents ability to unilaterally explore dialogical options not currently present in a given protocol. The other unique advantage is that not only can an agent generate its expected moves given its model (e.g. norms, commitments,etc.) but it can also communicate its expectations for others. Whereas, these traditional agent-centric models typically only provide guidance for a single agent and depend on other agents also having the same model of communication to coordinate their conversation.

The details of dialogue structure theory is largely concerned with issues unique to human communication. Our focus on agent communication neatly avoids the most difficult issues associated with this research. DST has been useful for developing metaphors for the development of protocols and protocol synthesis, but its use is superficial. DST, whether used for annotating human dialogue or generating natural language, must concern itself with the minutia and subtleties that software communication does not. All aspects of agent communication is engineered. As a result, there is a regularity, simplicity, and explicitness to it. This artifactual form of communication is not complicated by thousands of years of culture and tradition that complicates human discourse (e.g. [14]). Having been saved from the most onerous tasks of DST, we are freed to concentrate on the much more modest task at hand which is using some basic ideas from the field to drive protocol synthesis.

3 The Protocol Language

Figure 1 defines the syntax of a protocol language taken from [15] which also gives a fuller explanation of the language and framework. The protocol consists a set of agent clauses, $A^{\{n\}}$. These clauses are defined by an agent definition made up of a role (*R*) and unique identification(*Id*). A role is defined in a similar way as Electronic Institutions: It is a way of defining communicative activity for a group of agents rather than individuals. This agent definition is expanded by a number of operations.

Operations can be classified in three ways: actions, control flow,and conditionals. Actions are the sending or receiving of messages,a no op, or the adoption of a role. Control Flow operations temporally order the individual actions. Actions can be put in sequence (one action must occur before the other), or given a choice point (one and only one action should occur before any further action). The '$\Rightarrow$' and '$\Leftarrow$' denote messages,*M*, being sent and received. On the left-hand side of the double arrow is the message and on the right-hand side is the other agent involved in the interaction.

Constraints can fortify or clarify semantics of the protocols. Those occurring on the left of the '$\leftarrow$' are postconditions and those occurring on the right are preconditions. The symbol ψrepresents a first order propositions. For example, an agent receiving a protocol with the constraint to believe a proposition$s$ upon being informed of s can infer that the agent ending the protocol has a particular semantic interpretation of the act of informing other agents of propositions.

The message passed between two agents using the protocol consists of three parts. The first is the actual message (*M*) the agent is wishing to express. The second is the full

$$\begin{array}{lll}
\mathcal{P} \in \text{Protocol} & ::= A^{\{n\}} & \\
A \in \text{Agent Clause} & ::= \theta :: op. & \\
\theta \in \text{Agent Definition} & ::= \textbf{agent}(R,Id) & \\
op \in \text{Operation} & ::= \text{no op} & \\
 & \mid\ \theta & \\
 & \mid\ (op) & \text{(Precedence)} \\
 & \mid\ M \Rightarrow \theta & \text{(Send)} \\
 & \mid\ M \Leftarrow \theta & \text{(Receive)} \\
 & \mid\ op1\ \textbf{then}\ op2 & \text{(Sequence)} \\
 & \mid\ op1\ \textbf{or}\ op2 & \text{(Choice)} \\
 & \mid\ M \Rightarrow \theta \leftarrow \psi & \text{(Prerequisite)} \\
 & \mid\ \psi \leftarrow M \Leftarrow \theta & \text{(Consequence)} \\
\rho \in \text{message} & ::= \langle M,\mathcal{P},\mathcal{P}'\rangle &
\end{array}$$

Fig. 1. An Abstract Syntax of the Protocol Language

protocol ($\mathcal{P}$)itself. This will be necessary for the dissemination of the protocol as new agents enter the system. ($\mathcal{P}$') is the current dialogue state. This is the set of agent clauses marked to show the progress of the dialogue and the current state of the interaction. The messages are marked as closed or failed depending on whether they are communicated successfully. Messages which have been communicated are encased by a 'c', *c(M)*. This explicit communication of the dialogue state provide a means of coordination.

It is possible to create an agent which retains no internal record of the state of the dialogue but rather uses the communicated dialogue state as a book mark for which to hold its place and remind it of the next communicative step it can take.

The engineering requirements for implementing this protocol language are relatively light-weight. Agents are required to share a dialogical framework. The same is required of Electronic Institutions, and is an unavoidable necessity in any meaningful agent communication. An agent must be able to understand the protocol, the dialogue state, and its role within the protocol.Agents need to be able to identify the agent clause which pertains to its function within the protocol and establish what actions it must take to continue the dialogue. This includes the ability to update the dialogue state to reflect any actions it chooses to perform. The are several examples of frameworks which use this protocol language [16, 17].

This protocol language is well suited for our purposes. By distributing the protocol during the interaction, the agents have providence over the interaction protocol allowing agents to make transformations. The explicit transmission of the dialogue state records and communicates the choices made as the protocol is realised. It also able to catalogue the transformations made and the resulting properties which now hold because of those changes. This allows the mechanism for the protocol synthesis we seek. Now, that transformations are possible it is important to ensure they are controlled and meaningful.

4 Transformations

There are various structures which occur in human dialogue which have a different semantic interpretation but share the same syntactical shape. For example, a question followed by an answer has the same structure as a statement and a confirmation. An agent sends a message which is followed by another message being received. It is therefore useful to generalise the vocabulary of transformations to those whose semantics can be uniquely identified by its syntactic structure. Otherwise a kind of semantic leakage occurs and ambiguity seeps into the dialogue and synthesis. The sort of distinctions of a question and answer versus a propose and accept should be dealt with at the ACL level. Our concern is makes no assumptions about the particular locutions used for the protocol.

In figure 2 we define a set of transformations.The number is restricted to all the valid syntactical amendments to the simplest protocol (the protocol of a single message being communicated). The process of pruning away errant transformations are shown in the unfortunately monstrous figure 3.

The library of transformations in figure 2 was created by taking all the possible permutations of the two message protocol given an atomic protocol- an atomic protocol being defined as a single message being communicated, as a more simpler (non-empty)

$$\begin{array}{lcl}
M1 \Rightarrow \theta & \xrightarrow{response(M1,M2)} & M1 \Rightarrow \theta \ \texttt{then}\ M2 \Leftarrow \theta \\
M1 \Rightarrow \theta & \xrightarrow{continuation(M1,R2)} & M1 \Rightarrow \theta \ \texttt{then}\ M2 \Rightarrow \theta \\
M1 \Rightarrow \theta & \xrightarrow{counter(M1,M2)} & M1 \Rightarrow \theta \ \texttt{or}\ M2 \Rightarrow \theta \\
\end{array}$$

Before a Message is Received

$$\begin{array}{lcl}
M1 \Leftarrow \theta & \xrightarrow{continuation(M1,M2)} & M1 \Leftarrow \theta \ \texttt{then}\ M2 \Leftarrow \theta \\
M1 \Leftarrow \theta & \xrightarrow{response(M1,M2)} & M1 \Leftarrow \theta \ \texttt{then}\ M2 \Rightarrow \theta \\
M1 \Leftarrow \theta & \xrightarrow{counter(M1,M2)} & M1 \Leftarrow \theta \ \texttt{or}\ M2 \Leftarrow \theta \\
\end{array}$$

Upon the Reception of a Message

$$c(M1 \Leftarrow \theta) \xrightarrow{clarification(M1,M2)} c(M1 \Leftarrow \theta) \ \texttt{then}\ M2 \Rightarrow \theta$$

Upon Failure of a Message

$$\begin{array}{lcl}
f(M1 \Rightarrow \theta) & \xrightarrow{correction(M1,M2)} & f(M1 \Rightarrow \theta) \ \texttt{then}\ M2 \Rightarrow \theta \\
f(M1 \Leftarrow \theta) & \xrightarrow{correction(M1,M2)} & f(M1 \Leftarrow \theta) \ \texttt{then}\ M2 \Rightarrow \theta \\
\end{array}$$

Fig. 2. The Vocabulary of Transformations

protocol cannot be conceived. Each of these single message protocols can be expanded to a two message protocol by the addition of an ' **then** ' or an ' **or** ' operator followed by another message either incoming or outgoing. The total number of these two message protocols is seventy-two. By excluding protocols not possible within the LCC framework, the set is thinned to twenty-four possible transformations shown at the top of 3. For example, a protocol cannot exist that has a closed message following an open one (e.g. $M1 \Leftarrow \theta$ **then** $c(M2 \Rightarrow \theta)$). This is because of the way the protocol is expanded by the LCC framework.

After the first pruning, the set of twenty-four sheds six:

$$\begin{array}{lcl}
M1 \Rightarrow \theta & \longrightarrow & M1 \Rightarrow \theta \ \texttt{or}\ M2 \Leftarrow \theta \\
M1 \Leftarrow \theta & \longrightarrow & M1 \Leftarrow \theta \ \texttt{or}\ M2 \Rightarrow \theta \\
f(M1 \Rightarrow \theta) & \longrightarrow & f(M1 \Rightarrow \theta) \ \texttt{or}\ M2 \Leftarrow \theta \\
c(M1 \Rightarrow \theta) & \longrightarrow & c(M1 \Rightarrow \theta) \ \texttt{or}\ M2 \Leftarrow \theta \\
f(M1 \Leftarrow \theta) & \longrightarrow & f(M1 \Leftarrow \theta) \ \texttt{or}\ M2 \Rightarrow \theta \\
c(M1 \Leftarrow \theta) & \longrightarrow & c(M1 \Leftarrow \theta) \ \texttt{or}\ M2 \Rightarrow \theta \\
\end{array}$$

Although not strictly illegal there will never be the ambiguity of whose turn it is to speak. This is due to the our limiting protocol synthesis to dialogues. Protocols for multiparty conversations could indeed have this ambiguity.

Failure is defined as both the inability to communicate at the semantic level (i.e. the message was sent and received but not sensible with respect to an agent's knowledge base) as well as the physical failure to send or receive a message. Either a message being received or a message being sent is considered failed. When the failed message is outgoing. The sending agent has marked the failure in the dialogue state (i.e That agent knows about the failure). The only possible transformation which should be applied is the sending of a second message, a *correction*. Another four transformations being cast away by this second pruning.

Before a Message is Sent	Before a Message is Received
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{or}\ M2 \Leftarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{or}\ M2 \Leftarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{or}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{or}\ M2 \Rightarrow \theta$
Upon Failure of a Sent Message	Upon Having Sent a Message
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta$
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta$
Upon Failure of a Received Message	Upon Having Received a Message
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta$
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta$

$\Downarrow$ `first pruning` $\Downarrow$

Before a Message is Sent	Before a Message is Received
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{or}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{or}\ M2 \Leftarrow \theta$
Upon Failure of a Sent Message	Upon Having Sent a Message
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta$
Upon Failure of a Received Message	Upon Having Received a Message
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta$

$\Downarrow$ `second pruning` $\Downarrow$

Before a Message is Sent	Before a Message is Received
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{or}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{or}\ M2 \Leftarrow \theta$
Upon Failure of a Sent Message	Upon Having Sent a Message
	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$
	$c(M1 \Rightarrow \theta) \longrightarrow c(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta$
Upon Failure of a Received Message	Upon Having Received a Message
	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta$
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$
	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta$

$\Downarrow$ `final pruning` $\Downarrow$

Before a Message is Sent	Before a Message is Received
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Leftarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{then}\ M2 \Rightarrow \theta$
$M1 \Rightarrow \theta \longrightarrow M1 \Rightarrow \theta\ \texttt{or}\ M2 \Rightarrow \theta$	$M1 \Leftarrow \theta \longrightarrow M1 \Leftarrow \theta\ \texttt{or}\ M2 \Leftarrow \theta$
Upon Failure of a Sent Message	Upon Having Sent a Message
$f(M1 \Rightarrow \theta) \longrightarrow f(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	
Upon Failure of a Received Message	Upon Having Received a Message
$f(M1 \Leftarrow \theta) \longrightarrow f(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$	$c(M1 \Leftarrow \theta) \longrightarrow c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta$

Fig. 3. Pruning the Possible Transformations

$$\begin{array}{lcl} f(M1 \Rightarrow \theta) & \longrightarrow & f(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta \\ f(M1 \Leftarrow \theta) & \longrightarrow & f(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta \\ f(M1 \Rightarrow \theta) & \longrightarrow & f(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta \\ f(M1 \Leftarrow \theta) & \longrightarrow & f(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta \end{array}$$

It is not possible to make a transformation on the closed atomic protocol of a single sent message. By the definition of LCC, the message has already been communicated and with it the protocol one wishes to transform. For this reason, we can dismiss any protocol synthesised upon a closed outgoing message such as these:

$$\begin{array}{lcl} c(M1 \Rightarrow \theta) & \longrightarrow & c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta \\ c(M1 \Rightarrow \theta) & \longrightarrow & c(M1 \Rightarrow \theta)\ \texttt{then}\ M2 \Rightarrow \theta \\ c(M1 \Rightarrow \theta) & \longrightarrow & c(M1 \Rightarrow \theta)\ \texttt{or}\ M2 \Rightarrow \theta \end{array}$$

This point, $c(M1 \Rightarrow \theta)$, in the dialogue state occurs after the agent has evaluated, made its decision with respect to the conversation, and has expanded the protocol and only just before the message with the protocol and the dialogue state are sent to the other agent. It would be too late to synthesise more protocol. The situation is different for an incoming message which has been closed. The agent has just received the message. It has marked the message as closed and is at the point to make a decision.

$$\begin{array}{lcl} c(M1 \Leftarrow \theta) & \longrightarrow & c(M1 \Leftarrow \theta)\ \texttt{then}\ M2 \Leftarrow \theta \\ c(M1 \Leftarrow \theta) & \longrightarrow & c(M1 \Leftarrow \theta)\ \texttt{or}\ M2 \Leftarrow \theta \end{array}$$

For a closed received message the only transformation which can be applied is the addition of an outgoing message. From this final pruning, five more transformations can be scratched from the list leaving a more manageable nine shown in figure 2. The bottom of figure 3 shows the exhaustive set of the only possible syntactic transformations from an atomic protocol to one with two steps. Given all the possible two step protocols, one can apply the transformations to each of those and have all the permutations of a three step protocol, and in turn apply the transformations again to have all four step protocols. This can be done indefinitely in order to represent all possible protocols used within the LCC framework. There is no universally acceptable model of conversation, but we can map phenomena identified in DST research to the identified transformations. This is not implying that the transformation is an exact match to all similar phenomena in human dialogue, but that the mapping have an easily identifiable similarity. This is how figure 2 is derived from the final nine in figure 3.

In dialogues, humans cue for response by a number of verbal and non-verbal cues. This is captured by the two transformations in figure 2. A message is sent and at some point later a message is received from the same agent. The messages and their content can be said to be a *response*.

During discussions, humans will provide choice to their dialogical partners when appropriate. This same need exists in agent communication. The *counter* transformation allows agents to introduce this type of step in dialogues. Here we have a departure from the phenomenon occurring in human dialogue, versus agent interaction protocols. Rarely in human conversations are the options for response so explicitly stated as in our example. In agent communication it is not only common, but usually necessary.

Another feature of human dialogues is the use of cues to signify they wish to continue their turn in the dialogue. The*continuation* transformation enables software agents to do the same. The protocol coordinates whose turn it is to speak and an agent wishing to communicate more than one locution would not need a signalling phrase usually required for polite human dialogue but instead have a protocol allowing the multiple messages to be communicated.

Clarifications and Corrections are of great interest to those studying dialogue structures [18, 10].Corrections are usually reactions to failures in the dialogue. We have addressed outright failures such as message loss or complete misunderstanding as criteria for a *correction* transformation. Whereas, *clarifications* occur when a message received is understood but found to be wanting in detail. An agent providing a date but the other agent needs a year for the date as well*clarification* versus an agent communicating a seemingly erroneous date such as the tenth day of the seventeenth month*correction*. The message encapsulated by a ‘c’ before the*clarification* transformation represents in the protocol language that the message has been sent. The ‘f’ encapsulation represents a message failure which is the requirement for an agent making a *correction* transformation.

The transformations described are as generic as the LCC framework.There is no assumption of the rational make up of agents, the ACL involved or the domain ontology. In order for the transformations to make sense for a particular domain, it is necessary to define specific instances of the dialogue structures with respect to the domain being discussed and the locutions being communicated. These instances serve as synthesis rules. The figure 6 shows the rules for an agent to produce the dialogical steps to play an information-seeking dialogue game. They dictate for the agent what is considered the all proper responses, counters,continuations, corrections or clarifications given the ACL and domain of the dialogue.

For example, in a *response* the protocol has two messages, one coming in and one going out, separated by the ‘ **then** ’ operator. The synthesis rules for the agent say just what locution can be used fora *response* transformation.

response(ask(X),tell(X)).

The synthesis rule above says that the proper *response* for an*ask* locution is a *tell* and their content is the same.Given this synthesis rule, if the agent, we’ll call him ‘a’, has a protocol which is just the sending of an ask to agent b, written as:

$$agent(Proposition, a) ::= ask(Proposition) \Rightarrow agent(,b).$$

Agent *a* can synthesise a two step protocol which provides the protocol step to allow agent b to respond.

$$\begin{aligned}&agent(Proposition, a) ::= ask(Proposition) \Rightarrow agent(,b)\\&\mathbf{then}\ tell(Proposition) \Leftarrow agent(,b).\end{aligned}$$

We take advantage of the common knowledge mechanism in LCC to communicate the synthesis rules. This provides a public declaration of the rules that synthesised the protocol and the ability for other agents to employ the rule for synthesising. We now turn to describe the process of synthesis.

5 Synthesising Protocols

The process of synthesis progresses forward upon the last synthesised message. This is to prevent transformations such as figure 4. The two responses are performed with respect the first message, *M1*. This could go on indefinitely as the agent repeatedly applies synthesis rules with respect to *M1*.Similarly in human dialogue it is not possible to unsay what has been said. It is only possible to correct what has been said afterward.

$$\begin{array}{lcl}
\mathit{M1} \Rightarrow \theta & \xrightarrow{response(\mathit{M1},\mathit{M2})} & \mathit{M1} \Rightarrow \theta \; \texttt{then} \\
 & & \mathit{M2} \Leftarrow \theta \\
 & & \\
\mathit{M1} \Rightarrow \theta \; \texttt{then} & \xrightarrow{response(\mathit{M1},\mathit{M3})} & \mathit{M1} \Rightarrow \theta \; \texttt{then} \\
\mathit{M2} \Leftarrow \theta & & \mathit{M3} \Leftarrow \theta \; \texttt{then} \\
 & & \mathit{M2} \Leftarrow \theta
\end{array}$$

Fig. 4. An illegal Transformation

This is avoided by stepping forward to the last step of the protocol synthesised, and evaluating whether there are any synthesis rules to apply for that message. This way the protocol continues to expand but only in one direction, forward. The synthesis engine avoids state explosion and replication by halting further synthesis after the use of a *counter* rule. Until by communication or the agents choosing a single path exists, the engine cannot continue to synthesise the protocol.

LCC deals with meta-dialogical (e.g. deontic) concerns in a number of ways, one of which is the use of constraints. The use of constraints also deals context-dependent dialogical issues. The use of constraints with the synthesis rules also provide this functionality. For example, a synthesis rule can be written like this:

$$\textbf{response(}ask(X), tell(X) \;\leftarrow\; hasPrivileges(X, \theta)\textbf{)}.$$

This could be described as the proper response to an *ask*about 'X' is a *tell* about 'X' but only if the agent θhas privileges to that information. The synthesis engine puts the constraint in the appropriate agent clause in accordance with the syntactical rules of LCC. By the definition of LCC, the construction of a constraint on the left hand side of the $\leftarrow$ may only exists upon a received message (e.g. $\mathit{M1} \Leftarrow \theta$) and having a constraint on the right hand side is for outgoing messages (e.g.$\mathit{M1} \Rightarrow \theta$). Since this is a case, it is unambiguous for the synthesis engine to place the message and constraint onto the correct agent's clause.

In practice the agent will have a set of synthesis rules. Figure 6 shows the set of synthesis rules which enables an agent to synthesise a protocol similar to the one defined in [19].

This set of rule consists of several *responses* and*counters*. The last step of the synthesised protocol and the content of the locution informs which rules can be applied. Additionally, further constraints can be defined on synthesis rules to restrict their application. This is subtlety different than constraining the occurrence of a message in a synthesised protocol which is shown happening in rules*f*, *g*, and *h*.

6 An Example Using Dialogue Games for Synthesis

This section will use the Information-Seeking dialogue game similiar to [19][1] and show the use of synthesis rules provide versatility to the protocol based approach to dialogue. Figure 5 shows the information seeking game defined in the LCC protocol language. We will describe a set of synthesis rules which are sufficient to cover the conversation space defined by that protocol, but also show how to define stricter variations of the rules to reflect desirable properties different agents and their engineers might require. With this increasing strictness we shift the responsibility for protocol synthesis away from the agent to the rules themselves. I refer to this as the property of tolerance. A protocol is said to be more tolerant if it has a very liberal definition of the conversation space. In other words, there is a large number of possible paths the participants can take and still be within the protocol. The least tolerant protocol are the more orthodox static ones commonly used in agent communication. There is a strict ordering of messages to be exchanged as well as norms to which must be adhered. Agents using these intolerant protocols have little freedom, but the conversation will progress with increased reliability. The versatility of LCC synthesis attempts to address protocols along this spectrum.

Figure 6 shows our set of synthesis rules which can reproduce the dialogue game protocol in figure 5. The lines of code in the protocol are numbered. Although we will step through the protocol from the perspective of the initiator of the dialogue (i.e. the seeker) the protocol synthesised produces the symmetric clause for the dialogical partner. Lines 2 and 3 of figure 5 are captured by rule *a* in figure 6. The synthesis rule of *response*produces a message going out (question(P)) followed by a message coming in (assert(P)) separated by the operator ‘ `then` ’. According to the protocol an agent could also respond with an *assert* of the negation as well as asserting *unknown* (lines 4 and 5). A protocol can also be synthesised with these steps with rules*b* and *c*. In the original protocol the assertion of *unknown* ended the conversation. This strictness is not preserved the response rules of *d*. A more specific rule could have been defined disallow this step such as:

$$d')\ \textbf{response(}assert(R), accept(R)\ \leftarrow\ R\ \neq \textit{unknown}\textbf{)}.$$

Without the constraint, the synthesis' laxity is due to the uniqueness and transitivity of variables in the synthesis rules.This uniqueness is because the variables in the individual rules only refer to the same variable within that rule. Their scope does not extend beyond that rule. The *P* in rule *a* is not the same *P* as in rule *b*. What does make them the same is if the rules are applied iteratively and the *assert(P)* of the second part of rule *a* is the *assert(P)* of the first part of rule *b*. By transitivity *P* becomes the same through out both transformations. This is why one could apply rule *d* to an assert(unknown) message. Such flexibility puts the burden on the agent to not perform such an operation if it is deemed to be prohibited.

Not only can an agent *accept* an assertion but it should be able to *challenge* one. Rule *e* enables that. Rules*f*, *g*, and *h* provide an example of how to constrain the transformations given some condition in the conversation. An agent can respond to a challenge

[1] The only difference is the inclusion of acknowledgements to all propositions accepted.

```
$a(seeker(P,B),A)$ ::=
  $question(P) \Rightarrow a(provider(P,A),B)$ then
    $(assert(P) \Leftarrow a(provider(P,A),B)$ then $a(challenger([P],B),A))$ or
    $(assert(not(P)) \Leftarrow a(provider(P,A),B)$ then $a(challenger([not(P)],B),A))$ or
    $assert(unknown) \Leftarrow a(provider(P,A),B)$.

$a(provider(P,A),B)$ ::=
  $question(P) \Leftarrow a(seeker(P,B),A)$ then
    $(assert(P) \Rightarrow a(seeker(P,B),A)$ then $a(defender([P],A),B))$ or
    $(assert(not(P)) \Rightarrow a(seeker(P,B),A)$ then $a(defender([not(P)],A),B))$ or
    $assert(unknown) \Rightarrow a(seeker(P,B),A)$.

$a(challenger(List,B),A)$ ::=
  $null \leftarrow List = []$ or
  $((accept(R) \Rightarrow a(defender(List,A),B) \leftarrow List = [R|T]$ then
    $ack(R) \Leftarrow a(defender(List,A),B)$ then
      $a(challenger(T,B),A))$ or
  $(challenge(R) \Rightarrow a(defender(List,A),B) \leftarrow List = [R|T]$ then
    $assert(S) \Leftarrow a(defender(List,A),B)$ then
      $a(challenger(S,B),A)$ then $a(challenger(T,B),A)))$.

$a(defender(List,A),B)$ ::=
  $null \leftarrow List = []$ or
  $((List = [R|T] \leftarrow accept(R) \Leftarrow a(challenger(List,B),A)$ then
    $ack(R) \Rightarrow a(challenger(List,B),A)$ then
      $a(defender(T,A),B))$ or
  $(List = [R|T] \leftarrow challenge(R) \Leftarrow a(challenger(List,B),A)$ then
    $assert(S) \Rightarrow a(challenger(List,B),A) \leftarrow support(R,S)$ then
      $a(defender(S,A),B)$ then $a(defender(T,A),B)))$.
```

Fig. 5. A Protocol for a Information Seeking Dialogue Game

a) **response**($question(P), assert(P)$).
b) **counter**($assert(P), assert(not(P))$).
c) **counter**($assert(not(P)), assert(unknown)$).
d) **response**($assert(R), accept(R)$).
e) **counter**($accept(R), challenge(R)$).
f) **response**($challenge(R), assert(S) \leftarrow support(R,S)$).
g) **response**($assert(S), accept(R) \leftarrow memberOf(R,S)$).
h) **counter**($accept(R), challenge(R) \leftarrow memberOf(R,S)$).
i) **response**($accept(P), ack(R)$).
j) **response**($ack(P), accept(R)$).

Fig. 6. Synthesis Rules for an Information Seeking Game

with an assertion of the grounds for the argument as long as it can satisfy the constraint that those grounds are the support for the proposition that is challenged. Rules *d* and *e* deal with the correct responses for a single proposition. Rules *g* and*h* deal with the responses to a set of propositions with the added constraint that proposition that is accepted or challenged is a member of that set of propositions. These rules differ from rules*d* and *e* because the constraint ensures *S* is a list rather than a single proposition.

The final two rules give the ability to respond to all the propositions under consideration. A proper response to an acknowledgement is the acceptance of some other proposition. The*accept* is subject to the counter rule *e* which enables the agent to consider all the supporting arguments. In the protocol this was done through recursion enabled by the use of roles. Synthesis is driven by the locutions and their relationships and as such does not have some encapsulating data structure that can force iteration over the set of supporting arguments. The synthesised protocols produced are much more tolerant than the original and depend on the discretion of the agent doing the synthesis.

```
$a(seeker(\rho, agentB), agentA) ::=$
$c(question(\rho) \Rightarrow a(provider(\rho, agentA), agentB))$ then
$c(assert(\rho) \Leftarrow a(provider(\rho, agentA), agentB))$ then
  $a(challenger([\rho], agentB), agentA) ::=$
  $c(challenge(\rho) \Rightarrow a(defender([\rho], agentA), agentB))$ then
  $c(assert([\alpha, \beta, \gamma]) \Leftarrow a(defender([\rho], agentA), agentB))$ then
    $a(challenger([\alpha, \beta, \gamma], agentB), agentA) ::=$
    $c(accept(\alpha) \Rightarrow a(defender([\alpha, \beta, \gamma], agentA), agentB))$ then
    $c(ack(\alpha) \Leftarrow a(defender([\alpha, \beta, \gamma], agentA), agentB))$ then
      $a(challenger([\beta, \gamma], agentB), agentA) ::=$
      $c(accept(\beta) \Rightarrow a(defender([\beta, \gamma], agentA), agentB))$ then
      $c(ack(\beta) \Leftarrow a(defender([\beta, \gamma], agentA), agentB))$ then
        $a(challenger([\gamma], agentB), agentA) ::=$
        $c(accept(\gamma) \Rightarrow a(defender([\gamma], agentA), agentB))$ then
        $c(ack(\gamma) \Leftarrow a(defender([\gamma], agentA), agentB))$ then
      $a(challenger([], agentB), agentA)))$ then
    $a(challenger([], agentB), agentA).$
```

Fig. 7. Resulting Dialogue State using the Information Seeking Protocol

Figure 7 shows the resulting dialogue state for the initiating agent of the information seeking dialogue game using the protocol of figure 5. The various alternative messages (i.e. the **or** branches not taken) do not appear as the dialogue state only shows the choices made during the conversation. In this example the agent questioned the proposition ρ. The other agent replied with its assertion. The first agent challenged the assertion to which it received the reply of the set of the propositionsα,β, and γ. This set being the support for the original proposition. Obligingly, the agent accepts all supporting propositions for ρ and the other agent returns the courtesy by acknowledging each acceptance in turn.

Figure 8 is the process and the construction of the same instance of the information seeking dialogue game. Rather than using the prefabricated dialogue game protocol, the agent constructs the game during the interaction as defined by the synthesis rules of figure 6. At step one, the agent applies the synthesis rules *a, b,* and *c* stopping after the*counter* rules. Nothing has been communicated yet and the now synthesised protocol resembles the conversational choices provided by the *seeker* and *provider* roles. One agent can ask a question and the other can reply with an assertion, an assertion of the negation, or an assertion of *unknown*. In our example, two messages are then passed, question(ρ) and assert(ρ). These messages are recorded as closed in the dialogue state and the alternative locution choices are no longer shown.

For step two, the rules *d* and *e* are applied allowing the agent to either accept or challenge the other agent's assertion.This is the same behaviour allowed by the adoption of the*challenger* role.

In step three, the agent chooses to challenge but before the message is sent the rules *f, g,* and *h* are applied. This provides *agentB* with the ability to assert the supporting propositions, α, β, and γ and for *agentA*himself to respond with an acceptance or challenge upon an element of that proposition set. The current synthesis rules depend on the

Rule *a*, *b*, and *c* applied

(1) $question(\rho) \Rightarrow a(\_, agentB)$ **then** $assert(\rho) \Leftarrow a(\_, agentB)$ **or**
$assert(not(\rho)) \Leftarrow a(\_, agentB)$ **or** $assert(unknown) \Leftarrow a(\_, agentB)$

Rule *d* and *e* applied

(2) $c(question(\rho)) \Rightarrow a(\_, agentB))$ **then** $c(assert(\rho) \Leftarrow a(\_, agentB))$ **then**
$accept(\rho) \Rightarrow a(\_, agentB))$ **or** $challenge(\rho) \Rightarrow a(\_, agentB)$

Rule *f*, *g*, and *h* applied

(3) $c(question(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert(\rho) \Leftarrow a(\_, agentB))$ **then**
$c(challenge(\rho) \Rightarrow a(\_, agentB))$ **then** $assert([\alpha, \beta, \gamma]) \Leftarrow a(\_, agentB)$ **then**
$accept(\alpha) \Rightarrow a(\_, agentB)$ **or** $challenge(\alpha) \Rightarrow a(\_, agentB)$

Rule *i*, *j*, and *e* applied

(4) $c(question(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert(\rho) \Leftarrow a(\_, agentB))$ **then**
$c(challenge(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert([\alpha, \beta, \gamma]) \Leftarrow a(\_, agentB))$ **then**
$accept(\alpha) \Rightarrow a(\_, agentB)$ **then** $ack(\alpha) \Leftarrow a(\_, agentB)$ **then**
$accept(\beta) \Rightarrow a(\_, agentB)$ **or** $challenge(\beta) \Rightarrow a(\_, agentB)$

Rule *i*, *j*, and *e* applied

(5) $c(question(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert(\rho) \Leftarrow a(\_, agentB))$ **then**
$c(challenge(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert([\alpha, \beta, \gamma]) \Leftarrow a(\_, agentB))$ **then**
$c(accept(\alpha) \Rightarrow a(\_, agentB))$ **then** $c(ack(\alpha) \Leftarrow a(\_, agentB))$ **then**
$accept(\beta) \Rightarrow a(\_, agentB)$ **then** $ack(\beta) \Leftarrow a(\_, agentB)$ **then**
$accept(\gamma) \Rightarrow a(\_, agentB)$ **or** $challenge(\gamma) \Rightarrow a(\_, agentB)$

Rule *i*, *j*, and *e* applied

(6) $c(question(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert(\rho) \Leftarrow a(\_, agentB))$ **then**
$c(challenge(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert([\alpha, \beta, \gamma]) \Leftarrow a(\_, agentB))$ **then**
$c(accept(\alpha) \Rightarrow a(\_, agentB))$ **then** $c(ack(\alpha) \Leftarrow a(\_, agentB))$ **then**
$c(accept(\beta) \Rightarrow a(\_, agentB))$ **then** $c(ack(\beta) \Leftarrow a(\_, agentB))$ **then**
$accept(\gamma) \Rightarrow a(\_, agentB)$ **then** $ack(\gamma) \Leftarrow a(\_, agentB)$

No more protocol is synthesised

(7) $c(question(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert(\rho) \Leftarrow a(\_, agentB))$ **then**
$c(challenge(\rho) \Rightarrow a(\_, agentB))$ **then** $c(assert([\alpha, \beta, \gamma]) \Leftarrow a(\_, agentB))$ **then**
$c(accept(\alpha) \Rightarrow a(\_, agentB))$ **then** $c(ack(\alpha) \Leftarrow a(\_, agentB))$ **then**
$c(accept(\beta) \Rightarrow a(\_, agentB))$ **then** $c(ack(\beta) \Leftarrow a(\_, agentB))$ **then**
$c(accept(\gamma) \Rightarrow a(\_, agentB))$ **then** $c(ack(\gamma) \Leftarrow a(\_, agentB))$

Fig. 8. Synthesis and Expansion of the Same Information Seeking Dialogue Game

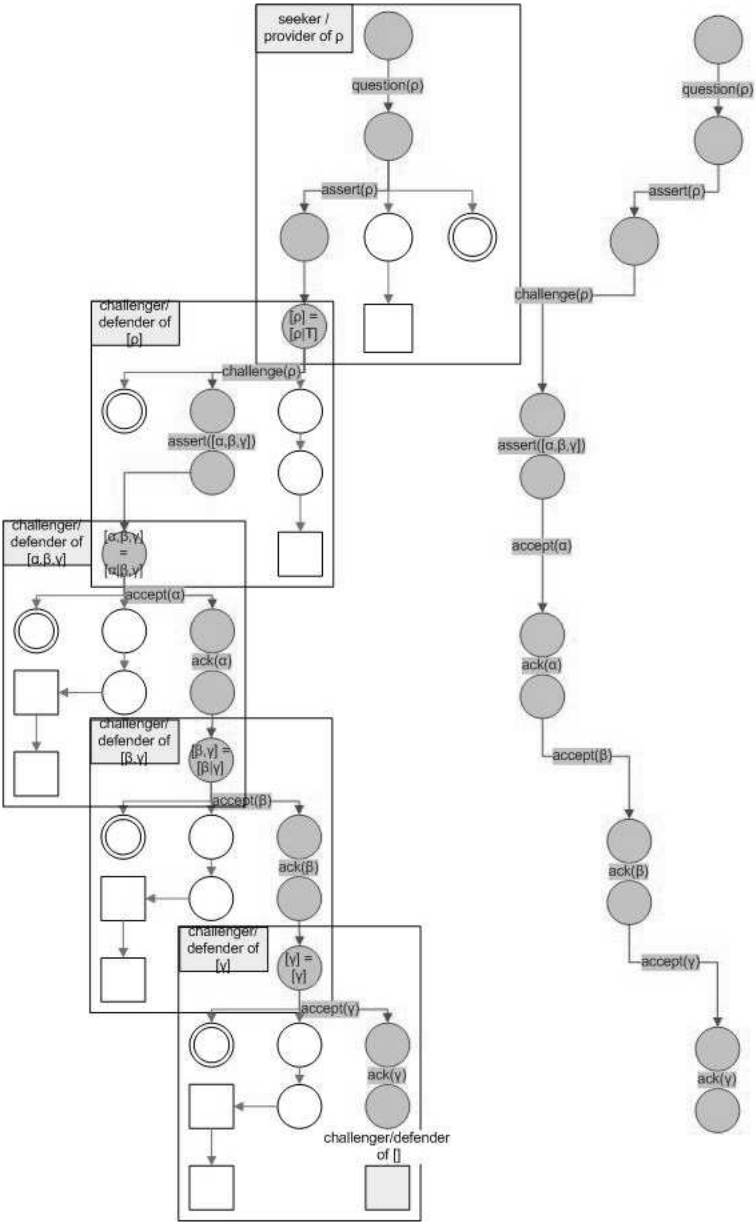


Fig. 9. Two Versions of the Dialogue State

agent to decide which proposition to consider for acceptance or not,whereas the protocol of figure 5 gave the agent no choice and ensured that all propositions are considered.

Step four occurs after the challenge is sent and the assertion of the support is received. The rules *i, j,* and *e* are used to allow the agent to acknowledge the accept as well as acceptor challenge one of the other propositions of the support. Like the dialogue in figure 7, step five, six and seven repeatedly use the rules *i, j,* and *e* to assert each supporting proposition which the agents in turn accept and acknowledge.

Figure 9 shows a graphical representation of the final dialogue state. The protocolled approach is on the left and the synthesised approach is on the right. On the left, the boxes represent the roles that the locutions are defined within. They are absent in the synthesis as there is no explicit representation of roles used. The figure shows that the dialogue that occurred is the same only the means of its production differed.

If necessary by the addition of one more rule we can have the ability to embed the information seeking dialogue games such as the complex dialogue games described in [20]. This allows more complex dialogue games consisting of more than one instance of an information seeking game. It simply requires an additional synthesis rule.

k) **response(**_,question(P)**)**.

By allowing the *response* to any message to be the first message of the information seeking game (i.e. the commencement rule), an agent can initiate that type of game at any point within another.

7 Conclusions

The use of LCC and the framework provides agents with the ability to communicate their interaction protocols as well as coordinate their own dialogues. Once agents are given this control over their interactions and the social norms as defined by the protocol the possibility of modifying the interaction protocol to address runtime needs can be explored. Initial work on this idea was reported in [21]. These papers described the process of making transformations to existing protocols by inserting and deleting portions of protocol. Though this approach worked there was a problem with traceability of the transformations. There was no simple way to identify where and when transformations occurred once the dialogue had ended. By the use of the synthesis rules, the user or agent can trace the construction of the protocol (i.e. given a set of synthesis rules one can construct a given protocol and vise versa).

With the exhaustive set of syntactic transformations, an agent may synthesise any dialogue protocol that can be defined in LCC including the use of constraints. When these transformations are defined by a set of synthesis rules using domain specific knowledge,the agent can synthesise a 'just in time' protocol to dynamically explore the conversation space. The protocol is constructed given the present dialogue state rather than the use of a static protocol which had been defined a priori. This is desirable when it is impossible or ungainly to define a protocol beforehand to address all possible paths through the conversation space.

It is recognised that the use of a distributed protocol and allowing agents to modify it during run time presents unique challenges.There are issues such as trust, consensus, writing privileges,etcetera. It is also recognised that the use of protocol synthesis would not be practical for domains with a high degree of uniformity and regularity in its communication. Plain LCC would be a better choice. Although, synthesis could be used for automatic protocol construct. The agent could initially use synthesis but later employ the constructed protocol rather than repeatedly synthesising the same protocol. We have not fully begun to explore this. Currently,its use has been restricted to systems where the dialogue is driven by the messages which occur at execution and allowing the agent to react by applying the appropriate synthesis rule to construct more protocol steps. This is an advantage to traditional agent-centric communication where the agent reasons about which message is should be sent. By exploiting the distribute protocol framework, it constrains the number of communicative actions which can be preformed for itself and its dialogical partners.

References

1. Aristotle: Topics. Clarendon Press (1997)
2. Georgeff, M., Pell, B., Pollack, M., Tambe, M., Wooldridge, M.: The belief-desire-intention model of agency. In Müller, J., Singh, M.P., Rao, A.S., eds.: Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98). Volume 1555., Springer-Verlag: Heidelberg, Germany (1999) 1–10
3. Finin, T., Fritzson, R., McKay, D., McEntire, R.: KQML as an Agent Communication Language. In Adam, N., Bhargava, B., Yesha, Y., eds.: Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, MD, USA, ACM Press (1994) 456–463
4. Singh, M.P.: A social semantics for agent communication languages. In Dignum, F., Greaves, M., eds.: Issues in Agent Communication. Springer-Verlag: Heidelberg, Germany (2000) 31–45
5. Amgoud, L., Maudet, N., Parsons, S.: Modeling dialogues using argumentation. In: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000), IEEE Computer Society (2000) 31
6. Bratman, M.: Intention, Plans, and Practical Reason. Havard University Press (1987)
7. Searle, J.: Speech Acts. Cambridge University Press (1969)
8. Walton, D., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. SUNY press, Albany, NY, USA (1995)
9. Core, M.G., Allen, J.F.: Coding dialogues with the DAMSL annotation scheme. In Traum, D., ed.: Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines, Menlo Park, California, American Association for Artificial Intelligence (1997) 28–35
10. Asher, N., Gillies, A.: Common ground, corrections and coordination. Argumentation **17** (2003) 481–512
11. Searle, J.: (on) Searle on Communication. Cambridge University Press (1969)
12. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In van Hoe, R., ed.: Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands (1996)
13. Rovatsos, M.: Computational Interaction Frames. PhD thesis, Department of Informatics, Technical University of Munich (2004)
14. Rickard, P.: A History of the French Language. Routledge (UK) (1989)
15. Robertson, D.: Multi-agent coordination as distributed logic programming. In: Proceedings for International Conference on Logic Programming. (2004)
16. Lambert, D., Robertson, D.: Matchmaking multi-party interactions using historical performance data. In: AAMAS. (2005) 611–617

17. Hassan, F., Robertson, D., Walton, C.: Addressing constraint failures in an agent interaction protocol. In: In Proceedings of the 8th Pacific Rim International Workshop on Multi-Agent Systems, Kuala Lumpur (2005)
18. Ginzburg, J.: Dynamics and the semantics of dialogue. In Seligman, J., Westerståhl, D., eds.: Logic, Language, and Computation. CSLI, Stanford, Ca (1996) 221–237
19. Parsons, S., Wooldridge, M., Amgoud, L.: An analysis of formal inter-agent dialogues. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems, ACM Press (2002) 394–401
20. McBurney, P., Parsons, S.: Games that agents play: A formal framework for dialogues between autonomous agents. Journal of Logic, Language and Information **11** (2002) 315–334
21. McGinnis, J., Robertson, D.: Realizing agent dialogues with distributed protocols. In: Developments in Agent Communication. Volume 3396 of LNAI. Springer-Verlag (2004)

An Argumentation-Based Model for Reasoning About Coalition Structures

Leila Amgoud

Institut de Recherche en Informatique de Toulouse (IRIT)
118, route de Narbonne,
31062 Toulouse Cedex 4 France
amgoud@irit.fr

Abstract. Autonomous agents working in multi-agent environments need to cooperate in order to fulfill tasks. Generally, an agent cannot perform a task alone and needs help from the other agents. One of the solutions to this problem is to look for groups of agents which are able to perform the desired tasks better. Different algorithms have then been proposed for the task allocation via coalition formation. This last is generally seen as a two steps process: i) constructing the different solutions (called coalitions structures), then ii) discussing these solutions between the agents in order to select the best ones which will be adopted.

This paper provides a unified formal framework for constructing the coalitions structures. In fact, we will show that the problem of coalition formation can be defined in terms only of *a set of coalitions* whose structures are abstract, a *conflict* relationship between the coalitions and a preference relation between the coalitions. Three semantics for coalitions structures will be proposed: a basic semantics which will return a unique coalition structure, stable semantics and preferred semantics. These two last may return several coalitions structures at the same time. A proof theory of the basic semantics will also be proposed. The aim of this proof theory is to test whether a given coalition will be acceptable for the agent or not without computing the whole structure. We will show that this framework is general enough to capture different propositions made in the literature. An instantiation of our framework is given and its properties are studied.

1 Introduction

Generally, to perform complex tasks in multi-agent environments, agents need to coordinate either because tasks require many resources if they are to be performed by a single agent, or because certain tasks can be carried out more efficiently by specialized agents. One of the solutions to this problem is to look for groups of agents which are able to perform the desired tasks better. This means that agents may form *coalitions* which are temporary associations between agents in order to carry out joint tasks.

Coalition formation can be seen as a two steps process:

1. Generating the *coalition structures*. The idea here is to form the coalitions such that agents within a coalition should coordinate to achieve a task (or a set of tasks), but those in different coalitions do not.
2. Discussing these structures between the agents in order to select the best ones which will be adopted.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 217–228, 2006.

The way in which the coalitions structures are generated depend broadly on the type of the studied problem. In some applications, for example, it may be required that the tasks are independent. In some other applications, it may also be required that a single agent should belong only to one coalition at the same time.

Different algorithms have then been proposed for task allocation via coalition formation [1, 2, 7, 8, 11, 12, 6, 9, 10, 4], and consequently for generating coalition structures. Each of them tries to resolve a particular problem with particular constraints.

Inspired from work on argumentation theory, particularly the famous argumentation system developed in [5], this paper provides a *unified* and *general* formal framework for generating the coalitions structures. That framework is defined in terms of a *set of coalitions* considered as abstract entities, a *conflict* relationship between these coalitions and finally a *preference* relation between the coalitions. Three semantics of coalitions structures are given: the *basic* semantics which returns a unique coalitions structure, stable semantics and preferred semantics which are two different refinements of the basic one. These two last semantics may return several coalitions structures at the same time. We propose also a proof theory in the case of basis semantics. The idea here, is that instead of computing all the coalitions structure in order to know whether a given coalition is in that structure, we can check directly if it is a member of the structure or not. This gives a dynamic way for the agents to test the acceptability of a coalition. This work is of great importance since it allows agents to reason about the coalitions, and minimize the negotiation between agents in the second step of the coalition formation process. Moreover, this framework is general enough to capture different propositions made in the literature.

This paper is organized as follows: Section 2 presents the suggested abstract framework. Section 3 provides a proof theory testing whether a given coalition will be in the coalitions structure. 4 presents an instantiation of our framework. Section 5 is devoted to some concluding remarks and perspectives.

2 Formal Model for Generating Coalition Structures

The problem of task allocation via coalition formation can be defined as a finite set $\mathcal{N}$ of agents who should achieve a finite set $\mathcal{T}$ of tasks. Each agent aims at maximizing its own satisfaction and also the satisfaction of the whole multi-agent system in which it is a member. In the literature on coalition formation, each agent is supposed to be equipped with a function which returns its degree of satisfaction for each coalition.

A framework for generating coalition structures is defined as a triple consisting of a *set of coalitions*, a binary relation representing the *defeasibility* relation between these coalitions, and finally a *preference* relation between coalitions. Here, a coalition is an abstract entity whose role is only determined by its relation to other coalitions. Then its structure is not known. It may be, for instance, any subset of $\mathcal{N}$, or a subset of $\mathcal{N}$ which achieves a given task.

Regarding conflicts, these should capture the constraints imposed by the studied problem. For instance, if the considered application imposes that an agent belongs to a unique coalition, then two coalitions containing at least one common agent are conflicting.

The agents are able to *evaluate* each coalition. For instance, a coalition may have a *cost* and a *profit*. In such a case, one can imagine the value of a coalition as its profit minus its cost. The values of coalitions make it possible to compare them.

Thus, the coalition formation problem can then be represented as a four steps process:

1. constructing the coalitions.
2. defining the defeasibility and preference relations between these coalitions.
3. defining the acceptable coalitions. These will be the coalition structures.
4. concluding.

The three first steps consist of generating the coalitions structures, whereas the last one to discuss them between agents.

Definition 1 (Formal framework). *A* framework for generating coalition structures *(FGS) is a triplet $<\mathcal{C}, \mathcal{R}, \succ>$ where $\mathcal{C}$ is a set of coalitions, $\mathcal{R}$ is a binary relation representing a defeat relationship between coalitions, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$, and $\succ$ is a (partial or complete) preordering on $\mathcal{C}$.*

Definition 2. *A framework (FGS) is* finitary *iff for each coalition C there is a finite number of coalitions which defeat C.*

Different definitions for the defeat relation ($\mathcal{R}$) and for the preference relation ($\succ$) lead to different systems which may not return the same coalition structures.

Defeating coalition can in turn be defeated by other coalitions so we need to define a notion of the *status* of coalitions. Its definition takes as input the set of all possible coalitions and their mutual relations of defeat and preference, and produces as output a division of coalitions into three classes:

- The class $\underline{\mathcal{S}}_{\mathcal{R},\succ}$ of *acceptable coalitions*. They represent the coalition structure.
- The class $\mathcal{R}_{\mathcal{R},\succ}$ of *rejected coalitions*. They are those coalitions defeated by acceptable ones. Such coalitions will not belong to a coalition structure.
- Coalitions which are neither acceptable nor rejected are gathered in the so-called class of *coalitions in abeyance*: $Ab_{\mathcal{R},\succ} = \mathcal{C} \setminus (\underline{\mathcal{S}}_{\mathcal{R},\succ} \cup \mathcal{R}_{\mathcal{R},\succ})$.

Note that to define the rejected coalitions and the coalitions in abeyance of a given framework (FGS), we first need to determine the set of acceptable coalitions of that framework. Intuitively, it is clear that a coalition which is not defeated at all will be accepted and will belong to the coalition structure. In what follows, the set $\mathcal{C}_{\mathcal{R}}$ will gather all non-defeated coalitions.

Example 1. Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a FGS such that $\mathcal{C} = \{C_1, C_2, C_3, C_4, C_5\}$, $\mathcal{R} = \{(C_3, C_4), (C_4, C_3), (C_1, C_5)\}$ and $C_3 \succ C_4$, then $\mathcal{C}_{\mathcal{R}} = \{C_1, C_2\}$.

This notion of acceptability is not sufficient and is very restrictive. We refine it by accepting defeated coalitions provided that they are preferred to their defeaters. The idea here is to privilege strong coalitions.

Definition 3. *Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a FGS and C_1, C_2 be two coalitions of $\mathcal{C}$ such that C_1 $\mathcal{R}$ C_2. C_2* defends itself *against C_1 iff $C_2 \succ C_1$.*
A coalition defends itself iff it is preferred w.r.t $\succ$ to each of its defeaters.
$\mathcal{C}_{\mathcal{R},\succ}$ denotes the set of coalitions defending themselves against their defeaters.

Example 2. In example 1, since $C_3 \succ C_4$ then C_3 defends itself against C_4. Consequently, $\mathcal{C}_{\mathcal{R},\succ} = \{C_1, C_2, C_3\}$.

The set $\mathcal{C}_{\mathcal{R},\succ}$ contains also the coalitions which are not defeated (in the sense of the relation $\mathcal{R}$).

Property 1. Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a FGS. $\mathcal{C}_{\mathcal{R}} \subseteq \mathcal{C}_{\mathcal{R},\succ}$.

The set $\mathcal{C}_{\mathcal{R},\succ}$ is also too restricted since it discards coalitions which appear acceptable. Intuitively, if a coalition C_1 is less preferred than its defeater C_2 then it is weakened. But the defeater C_2 itself may be weakened by another coalition C_3 which defeats C_2 and is preferred to C_2. In this later case we would like to accept C_1 because it is defended by C_3. This corresponds to a joint defence point of view used in argumentation theory.

Definition 4. *Let $S \subseteq \mathcal{C}$. A coalition C_1 is* defended *by S iff $\forall$ $C_2 \in \mathcal{C}$, if C_2 $\mathcal{R}$ C_1 and not($C_1 \succ C_2$) then $\exists$ $C_3 \in S$ such that C_3 $\mathcal{R}$ C_2 and not($C_2 \succ C_3$).*

The coalition structure (i.e. the set of acceptable coalitions) is then characterized, by a *monotonic* function $\mathcal{F}$ that returns for each set of coalitions, the set of all coalitions that are defended by that set.

Definition 5. *Let $S \subseteq \mathcal{C}$. $\mathcal{F}(S) = \{C \in \mathcal{C} \mid C$ is defended by $S\}$.*

Since the function $\mathcal{F}$ is monotonic, the set of acceptable coalitions is defined as its least fixpoint. Moreover, when the framework FGS is finitary, the function $\mathcal{F}$ is *continuous* and then its least fixpoint can be obtained by iterative application of $\mathcal{F}$ to the empty set.

Definition 6 (Coalitions structure). *Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a finitary FGS. The coalitions structure is defined as:*

$$\underline{\mathcal{S}}_{\mathcal{R},\succ} = \bigcup \mathcal{F}^{i>0}(\emptyset)$$

By applying the characteristic function $\mathcal{F}$ to the empty set, we obtain exactly the set of coalitions defending themselves against their defeaters. More formally:

$$\mathcal{F}(\emptyset) = \mathcal{C}_{\mathcal{R},\succ}.$$

Thus,

$$\underline{\mathcal{S}}_{\mathcal{R},\succ} = \bigcup \mathcal{F}^{i>0}(\emptyset) = \mathcal{C}_{\mathcal{R},\succ} \cup [\bigcup \mathcal{F}^{i\geq 1}(\mathcal{C}_{\mathcal{R},\succ})]$$

The coalitions structure contains then the coalitions which defend themselves against their defeaters ($\mathcal{C}_{\mathcal{R},\succ}$) and also the coalitions which are defended (directly or indirectly) by coalitions of $\mathcal{C}_{\mathcal{R},\succ}$.

Before defining the proof theory, let's give some definitions.

Definition 7. *Let C_1, C_2 be two coalitions of $\mathcal{C}$, and $S \subseteq \mathcal{C}$.*

- C_1 attacks C_2 *iff C_1 $\mathcal{R}$ C_2 and not ($C_2 \succ C_1$).*
- C_1 disqualifies C_2 *iff* C_1 attacks C_2 *and not(*C_2 attacks C_1*)*
- *S* strictly defends *C_1 iff for all C_2 such that C_2 attacks C_1, then there is a $C_3 \in S$ such that C_3 disqualifies C_2.*

Theorem 1. $\forall\, C \in \underline{\mathcal{S}}_{\mathcal{R},\succ}$, $\underline{\mathcal{S}}_{\mathcal{R},\succ}$ *strictly defends* C.

In some cases, the set $\underline{\mathcal{S}}_{\mathcal{R},\succ}$ may be empty. Let's illustrate it on the following example.

Example 3. Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a FGS such that $\mathcal{C} = \{C_1, C_2\}$, $\mathcal{R} = \{(C_1, C_2), (C_2, C_1)\}$, $C_1 \succ C_2$ and $C_2 \succ C_1$. In this framework $\underline{\mathcal{S}}_{\mathcal{R},\succ} = \emptyset$.

In the above example, no structure is returned and consequently no coalition is formed. This is not always desirable in multi-agents applications. In order to palliate the limits of this notion of acceptability, we will consider other semantics defined mainly in [5] in argumentation context. Indeed, we will define *stable structures* and *preferred structures*. Unlike the above semantics of acceptability which returns only one coalition structure, these new semantics may generate several structures at the same time. Before presenting these semantics, let's first define a new notion of conflict-free:

Definition 8 (Conflict-free). $S \subseteq \mathcal{C}$. *S is conflict-free iff* $\nexists\, C_1, C_2 \in S$ *such that* $C_1\ \mathcal{R}\ C_2$ *and not* $(C_2 \succ C_1)$.

Definition 9 (Stable structures). *Let* $<\mathcal{C}, \mathcal{R}, \succ>$ *be a FGS, and* $S \subseteq \mathcal{C}$. *S is a* stable structure *iff*

1. *S is conflict-free.*
2. *S defeats any coalition which is not in S.*

Note that a framework FGS may have several stable structures. These stable structures correspond to different ways of achieving the tasks.

Example 4. In example 3, there are two stable structures $\underline{\mathcal{S}}_1 = \{C_1\}$ and $\underline{\mathcal{S}}_2 = \{C_2\}$.

Definition 10 (Preferred structures). *Let* $<\mathcal{C}, \mathcal{R}, \succ>$ *be a FGS, and* $S \subseteq \mathcal{C}$. *S is a* preferred extension *iff*

1. *S is conflict-free*
2. *S defends all its elements*
3. *S is maximal (for set inclusion) among the sets satisfying the 2 above conditions.*

Note that each framework FGS has at least one preferred structure.

Example 5. Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a FGS such that $\mathcal{C} = \{C_1, C_2, C_3, C_4, C_5\}$, $\mathcal{R} = \{(C_2, C_1), (C_2, C_5), (C_5, C_5), (C_4, C_2), (C_3, C_2), (C_3, C_4), (C_4, C_3)\}$, $C_2 \succ C_1$, C_5 and $C_4 \succ C_1$, and $C_3 \succ C_2$, and $C_3 \succ C_4$ and $C_4 \succ C_3$. In this framework $\underline{\mathcal{S}}_{\mathcal{R},\succ} = \emptyset$, whereas there are two preferred structures $\underline{\mathcal{S}}_1 = \{C_1, C_3\}$ and $\underline{\mathcal{S}}_2 = \{C_1, C_4\}$.

Property 2.
- Each stable structure is also a preferred one. However, the reverse is not always true.
- The coalition structure $\underline{\mathcal{S}}_{\mathcal{R},\succ}$ is included in every stable (resp. preferred) structure.

3 Proof Theory

So far, we have provided the semantics of a coalition structure by defining the set of acceptable coalitions it constrains, namely $\underline{\mathcal{S}}_{\mathcal{R},\succ}$. However, in practice we don't need to calculate the whole set $\underline{\mathcal{S}}_{\mathcal{R},\succ}$ in order to know the status of a given coalition. In this section we propose a test for membership for a coalition C, i.e. we propose a proof theory for testing whether C is in $\underline{\mathcal{S}}_{\mathcal{R},\succ}$ or not.

For that purpose, we are inspired by the work done in [3] in the context of argumentation theory.

The basic idea of this proof theory is to traverse the sequence $\mathcal{F}^1, ..., \mathcal{F}^n$ in reverse. Consider that C occurs for the first time in $\mathcal{F}^n$. We start with C, and then for any coalition B_i which attacks C, we find a coalition C_i in $\mathcal{F}^{n-1}$ which defends C. Now, because of Theorem 1, we are only interested in the strict defenders of a coalition, and the strict defenders of C will disqualify the B_i. The same process is repeated for each strict defender until there is no strict defender or defeater.

We can think of this process in terms of a dialogue game between two players P and O. P makes the coalition we are interested in and its defenders and the player O makes the counter-coalitions or defeaters.

Definition 11. *A* dialogue *is a nonempty sequence of moves,* $move_i$=$(Player_i, Coal_i)$ $(i \geq 0)$ *such that:*

1. $Player_i = P$ *iff* i *is even,* $Player_i = O$ *iff* i *is odd.*
2. $Player_0 = P$ *and* $Coal_0 = C$.
3. *If* $Player_i = Player_j = P$ *and* $i \neq j$ *then* $Coal_i \neq Coal_j$.
4. *If* $Player_i = P$*,* $i > 1$*, then* $Coal_i$ *disqualifies* $Coal_{i-1}$.
5. *If* $Player_i = O$ *then* $Coal_i$ *attacks* $Coal_{i-1}$.

A dialogue tree *is a finite tree where each branch is a dialogue.*

Example 6. Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a FGS such that $\mathcal{C}$ = $\{a_0, a_{01}, a_{02}, a_{10}, a_{11}, a_{12}\}$, $\mathcal{R}$ = $\{(a_{10}, a_0), (a_{01}, a_{10}), (a_{12}, a_{02}), (a_{02}, a_{10}), (a_{03}, a_{11}), (a_{11}, a_0)\}$. Let's suppose that $a_{03} \succ a_{11} \succ a_0$, $a_{01} \succ a_{10} \succ a_0$ and $a_{12} \succ a_{02}$, $a_{02} \succ a_{10}$. We are interested in the status of the coalition a_0. The corresponding dialogue tree is presented in Figure 1.

The dialogue tree can be considered as an AND/OR tree. A node corresponding to the player P is an AND node, and a node corresponding to the player O is an OR node. This

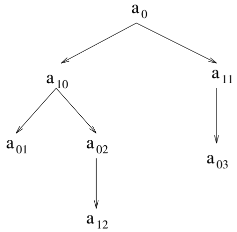


Fig. 1. A dialogue tree

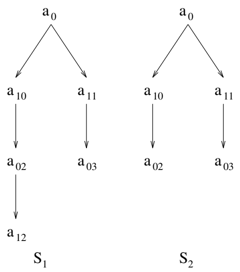


Fig. 2. Candidate sub-trees

is because a coalition is acceptable if it is defended against all its defeaters. The edges of a node containing a coalition of P represent defeaters so they all must be defeated. In contrast, the edges of a node containing a coalition of O represent defenders of P so it is sufficient that one of them defeats the coalition of O.

Definition 12. *A player* wins *a dialogue iff he makes the last coalition in the dialogue.*

A player who wins a dialogue does not necessarily win in all the sub-trees of the dialogue tree. To formalize the winning of a dialogue tree, the concept of a solution sub-tree is defined.

Definition 13. *A* candidate sub-tree *is a sub-tree of a dialogue tree containing all the edges of each AND node and exactly one edge of each OR node. A* solution sub-tree *is a candidate sub-tree whose branches are all won by P.*

Example 7. Thus the dialogue represented in example 3 has exactly two candidate sub-trees S_1 and S_2, Figure 2 .

Definition 14. P wins *a dialogue iff the corresponding dialogue tree has a solution sub-tree.*

Example 8. Thus P wins the dialogue presented in Figure 1 because S_2 is a solution sub-tree.

Definition 15. *Let $C \in \mathcal{C}$. A coalition C is* justified *iff there is a dialogue tree whose root is C, and which is won by the player P.*

Example 9. Thus the coalition a_0 is justified because the player P won the dialogue tree.

The main result from the proof theory is:

Theorem 2. *Let $<\mathcal{C}, \mathcal{R}, \succ>$ be a FGS.*

1. *$\forall\, C \in \mathcal{C}$, if C is justified then each coalition of P belonging to the solution sub-tree is in $\underline{\mathcal{S}}_{\mathcal{R},\succ}$, in particular C.*
2. *$\forall\, C \in \underline{\mathcal{S}}_{\mathcal{R},\succ}$, C is justified.*

In other words, the dialogue process constructs all acceptable coalitions, and only constructs acceptable coalitions and is thus sound and complete.

4 Application of the General Model

In order to illustrate our model, let's consider the problem of coalition formation described in [11]. The problem is that of task allocation among groups of autonomous agents. The idea is that given the set $\mathcal{T}$ of tasks, the system has as a whole to satisfy all the tasks or at least seek the satisfaction of as many tasks as possible, thus maximizing its benefit. In that work, a multi-agent system is supposed to perform a *service*. That service requires several *criteria* $<c_1, \ldots, c_r>$. For example, if the service is transportation, then the criteria will be $weight$, $size$, $volume$. Several *agents* $\mathcal{N} = \{a_1, \ldots, a_n\}$ are involved in this system. Each agent a_i is supposed to have a vector of non-negative *capabilities* $B^i = <b^i_1, \ldots, b^i_r>$. A capability b^i_j represents the capacity of the agent a_i regarding the criterion c_j. In the case of transportation, this means that an agent p has x weight, y size and z volume. The system has also a set $\mathcal{T} = \{t_1, \ldots, t_m\}$ of *tasks* to perform. To each task t a vector $B^t = <b^t_1, \ldots, b^t_r>$ of its capabilities is given. An element $b^{t_j}_k$ represents the amount of c_k necessary for its satisfaction.

Definition 16 (Agents system). *An agents system (AS) is a triple* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *such that:*

- $\mathcal{S} = <c_1, \ldots, c_r>$.
- $\mathcal{N} = \{a_1, \ldots, a_n\}$ *is a set of agents, where* $\forall\, a_i \in \mathcal{N}$ *there are:*
  1. *a vector* $B^i = <b^i_1, \ldots, b^i_r>$ *of its capabilities.*
  2. *a function* $Value$ *which returns the value of a given coalition.*
- $\mathcal{T} = \{t_1, \ldots, t_m\}$ *is a set of* tasks *to be performed, where* $\forall\, t \in \mathcal{T}$ *there is a vector* $B^t = <b^t_1, \ldots, b^t_r>$ *of the necessary capabilities for its achievement.*

A coalition is a group of agents that decided to work together in order to fulfill a given task.

Hypothesis 1. *We assume that:*

- *The tasks are independent.*
- *An agent cannot belong to more than one coalition at a time.*
- *A coalition can work on a single task at a time.*

4.1 The Notion of Coalition

A coalition is a group of agents that cooperate in order to achieve a common task. In fact, a coalition should be *minimal* since each coalition has a cost. And the more the coalition is large, the more costly it is. Moreover, an agent cannot be in a coalition if it is not useful and it cannot help in the achievement of the task. Before giving the formal definition of a coalition, let's first define formally when a task is achievable.

Definition 17. *Let* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *be an agents system,* $C \subseteq \mathcal{N}$ *and* $t \in \mathcal{T}$. *The group* C *of agents* can achieve *the task* t, *denoted by* $C \Vdash t$, *iff* $\forall\, 1 \leq j \leq r$, $\sum_{a_i \in C} b^i_j \geq b^t_j$.

The above definition says that a task is achievable by a group of agents if the capabilities of the agents taken together, are sufficient to what is required by the task.

We are now ready to define formally a coalition.

Definition 18 (Coalition). *Let* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *be an agents system. A coalition is a pair* $<C, t>$ *such that:*

1. $C \subseteq \mathcal{N}$
2. $t \in \mathcal{T}$
3. $C \Vdash t$
4. *C is minimal for set inclusion among the sets satisfying the above conditions.*

C will be called the support *of the coalition, and t its* task. *In what follows, $\mathcal{C}(AS)$ will denote the set of all the coalitions that can be built from* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$.

4.2 The Force of a Coalition

Each agent is supposed to be equipped with a function $Value$ which returns the value of a coalition according to the agent. The value of a coalition may be equal to the benefit obtained from the coalition minus the cost of that coalition. However, the value may be defined in different ways. For the sake of simplicity, we suppose that this value is given and it is a numerical value.

The values of coalitions make it possible to compare these coalitions. Indeed, the coalition with a greater value is stronger than the ones with a small value. Formally:

Definition 19. *Let C_1, $C_2 \in \mathcal{C}$. C_1 is more* beneficial *than C_2, denoted $C_1 \succ C_2$ iff* $Value(C_1) > Value(C_2)$.

4.3 Conflicts Between Coalitions

The *coalition structures* should satisfy the hypothesis already fixed when defining the problem. The first requirement is that an agent cannot belong to more than one coalition at the same time. Indeed, two coalitions, defined as shown in Definition 18 and containing at least one agent in common cannot be in the coalition structures. moreover, such coalitions are said to be conflicting. This kind of conflict will be called here “Interfere”. Formally:

Definition 20 (Interfering coalitions). *Let* $<C_1, t_1>$, $<C_2, t_2> \in \mathcal{C}(AS)$.
$<C_1, t_1>$ interferes *with* $<C_2, t_2>$ *iff* $C_1 \cap C_2 \neq \emptyset$.

Note that the above relation is symmetrical. The second requirement in the studied problem is that the same task cannot be affected to more than one coalition at the same time. In the coalition structure, it cannot then be the case that two coalitions achieve the same task. This requirement gives raise to another kind of conflict between coalitions. In what follows, this conflict will be called “Competition”. Formally:

Definition 21 (Competing coalitions). *Let* $<C_1, t_1>$, $<C_2, t_2> \in \mathcal{C}(AS)$. $<C_1, t_1>$ *is in* competition *with* $<C_2, t_2>$ *iff* $t_1 = t_2$.

The two above relations are brought together in a unique definition of defeat as follows:

Definition 22 (Defeat). *Let* $<C_1, t_1>$, $<C_2, t_2> \in \mathcal{C}(AS)$.
$<C_1, t_1>$ defeats $<C_2, t_2>$ *iff:*

- $<C_1, t_1>$ interferes *with* $<C_2, t_2>$ *or*
- $<C_1, t_1>$ *is in* competition *with* $<C_2, t_2>$.

4.4 Coalition Structures

Once the notions of coalition and defeasibility are defined, we are now able to introduce the system which will be used for generating the coalition structures.

Definition 23. *A* framework for generating coalition structures *is a triplet* $<\mathcal{C}(AS)$, $Defeat$, $\succ>$ *where* $\mathcal{C}(AS)$ *is the set of coalitions built from the agents system* $<\mathcal{S}$, $\mathcal{N}$, $\mathcal{T}>$ *using Definition 18,* $Defeat$ *is the relation given in Definition 22, and* $\succ$ *is a (partial or complete) preordering on* $\mathcal{C}(AS) \times \mathcal{C}(AS)$ *as shown in Definition 19.*

The coalition structure of this system is:

$$\underline{\mathcal{S}}_{Defeat,\succ} = \bigcup \mathcal{F}^{i>0}(\emptyset)$$

$$= \mathcal{C}_{Defeat,\succ} \cup [\bigcup \mathcal{F}^{i\geq 1}(\mathcal{C}_{Defeat,\succ})]$$

Let's define two functions: Supp and Task. The function Supp returns for a given set of coalitions, the set of all agents involved in that coalitions. The function Task returns for a given set of coalitions, the set of all tasks achievable by those coalitions.

Definition 24. *Let* $<\mathcal{C}(AS)$, $Defeat$, $\succ>$ *be a FGS. A coalition structure* S *is* complete *iff:*

1. *Supp(S) =* $\mathcal{N}$*, and*
2. *Task(S) =* $\mathcal{T}$*.*

The coalition structure $\underline{\mathcal{S}}_{Defeat,\succ}$ is not always complete as can be shown by the following example:

Example 10. Let $\mathcal{N} = \{a_1, a_2, a_3\}$ and $\mathcal{T} = \{t_1, t_2\}$. Let's suppose that the two following coalitions are constructed: $C_1 = <\{a_1, a_2\}, t_1>$ and $C_2 = <\{a_1, a_3\}, t_2>$. Suppose also that $C_1 \succ C_2$. The coalition structure contains only the coalition C_1. Thus, only agents a_1 and a_2 will participate for the achievement of a task. Moreover, only one task t_1 will be achieved.

The following result can be shown:

Theorem 3. *If the agents do not misrepresent the capabilities of the others, and if they have all the same values for the different coalitions, then their respective frameworks will all return the same coalition structure. Thus, there is no need to the negotiation step.*

This result is of great importance since it shows that with such a framework, more work is done by the agents themselves, and consequently this may minimize greatly the communication which is very costly.

5 Conclusion

This paper addresses the problem of coalition formation. The idea behind this problem is to form groups of agents able to perform more efficiently different services or tasks. The coalition formation process follows two steps: each agent constructs its coalition structures (i.e. the affectation of tasks to groups of agents), then it should discuss them with the other agents in order to reach an agreement on the structure which will be adopted.

Several proposals [2, 7, 8, 11, 12, 6, 9] have been introduced in the literature. The aim of these works is to present efficient algorithms for computing the coalition structures of each agent, and how the optimal solution can be reached by the agents within a short negotiation.

All the above works have in common the problem of coalition formation, however, each of them generally studies a particular application.

Inspired from works on argumentation theory, we have proposed a *unified*, *general* and *abstract* framework which allows to construct the coalition structures in an elegant way. The formal framework has three components: a set of coalitions, a defeasibility relation between the coalitions, and finally a preference relation between the coalitions. In this abstract framework, the notion of coalition remains an abstract entity whose role is only defined in terms of its relation with the other coalitions. The exact definition of a coalition depends broadly on the studied application. It may be any sub-set of $\mathcal{N}$ (the set of agents), and in this case the set of coalitions will be exactly $2^{\mathcal{N}}$. However, in some other application the definition of coalition is more precise. In the case of task allocation, a coalition is exactly a set of agents which are able to achieve together a given task (or a set of tasks).

Regarding the notion of defeasibility, it is induced and defined from the constraints of the application. Finally, the preference relation comes from the values that agents can assign to each coalition. The value of a coalition depends also on the application. It may represent the benefits, or the cost of a coalition.

We have proposed three semantics for the coalitions structures: the *basic structure* one which returns only one coalition structure, the *stable structures* and the *preferred structures*. The stable and preferred structures may return several solutions at the same time. Each of them corresponds to a particular point of view of the agent. However, these different coalition structures may not be equally preferred by the agent. For each coalition structure, an agent may have a value which is the sum of the values of the different coalitions in the structure.

Another important contribution of this paper is the proposed proof theory in the case of a basic structure. This proof theory aims at testing whether a given coalition will be in the coalition structure or not. This is very important since the agents are not obliged to compute all the coalition structure to know whether a coalition is good for them or not. This is particularly useful in the negotiation step. When an agent proposes a given coalition, the other agents can easily check its acceptability without computing the whole structure.

The framework has been instantiated in a particular application of task allocation. We have shown that if the agents don't misrepresent the capabilities of the other agents, and if the agents have the same preferences than there is no need to negotiate since their respective systems will return the same structure.

An extension of this work would be to study more deeply how the existing proposals fit in our framework, and to compare their results with the ones which will be given with our framework.

Another important extension of this work consists of studying the notion of preference relation. In this paper, this relation is very general, and as said before it reflects the importance of a coalition to an agent. However, in most agents' applications, the agents are autonomous and may have incomplete information about the environment and about the other agents of the system. Thus, we can easily imagine that an agent may construct coalitions which are more or less certain for it. Moreover, a coalition may satisfy more or less prioritized goals of an agent. It is then important to take these factors into account when comparing coalitions.

As already said, this work has been inspired from work on argumentation theory. In that particular context, several algorithms have been defined for computing the different semantics. An extension of this work would be to adapt those algorithms to coalition formation problem and to compare their complexities to the complexity of existing algorithms.

Finally, it is very important to study the impact of the proposed framework on the second step of a coalition formation problem.

References

1. S. Aknine, S. Pinson, and M. Shakun. Coalition formation methods for multi-agent coordination problems. *Group Decision and Negotiation*, July 2000.
2. S. Aknine, S. Pinson, and M. Shakun. Coalition formation problem: New multi-agent methods with preference models. In *Proceedings of the AAAI workshop on Coalition formation in dynamic multi-agent environments*, July 2002.
3. L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, Volume 34:197–216, 2002.
4. V. D. Dan and N. Jennings. Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of AAMAS'2004*.
5. 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.
6. M. Klush and A. Gerber. Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, pages 42–47, 2002.
7. S. Kraus, O. Shehory, and G. Tasse. The advantages of compromising in coalition formation with incomplete information. In *Proceedings of the AAMAS 2004*, pages 588–595.
8. S. Kraus, O. Shehory, and G. Tasse. Coalition formation with uncertain heterogeneous information. In *Proceedings of the AAMAS 2003*, pages 1–8.
9. T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case garantees. *Artificial Intelligence*, pages 209–238, 1999.
10. T. Sandholm and V. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, pages 99–137, 1997.
11. O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *Proceedings of IJCAI'1995*.
12. O. Shehory and S. Kraus. Methods for coalition formation task allocation via coalition formation. *Artificial Intelligence*, pages 165–200, 1998.

Argumentation-Based Multi-agent Dialogues for Deliberation

Yuqing Tang[1] and Simon Parsons[2]

[1] Department of Computer Science
Graduate Center, City University of New York
365 5th Avenue, New York, NY 10016, USA
ytang@gc.cuny.edu
[2] Department of Computer and Information Science
Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210 USA
parsons@sci.brooklyn.cuny.edu

Abstract. This paper presents an argumentation-based approach to deliberation, the process by which two or more agents reach a consensus on a course of action. The kind of deliberation that we are interested in is a process that combines both the selection of an overall goal, the reduction of this goal into sub-goals, and the formation of a plan to achieve the overall goal. We develop a mechanism for doing this, describe how this mechanism can be integrated into a system of argumentation to provide a sound and complete deliberation system, and show how the same process can be achieved through a multi-agent dialogue.

1 Introduction

Multi-agent planning is clearly an important topic for the field of multi-agents systems, and, as one might imagine, has been widely studied and for a long time. As [4] points out, there is a large variety of approaches, from distributed versions of classical AI planning techniques like NOAH [3] and partial planning [6], to techniques that were developed to exploit specific attributes of multi-agent systems like joint intentions [14,23], or the *intention-that* of SharedPlans [9]. Some of these approaches deal with multi-agent plans holistically [10], while others build plans for individual agents and then merge them [7]. Approaches as disparate as model checking [24] and auctions [26] have been adapted to generate multiagent plans.

In this paper we bring together aspects of multiagent planning and work in a field that has grown up more recently, argumentation-based dialogue [18]. While there has been much work on argumentation-based dialogue in the last few years—including that of Kraus [13], Maudet [15], McBurney [16], Reed [20], Schroeder *et al.* [21] and Sycara [22]—there is not yet a definitive account of what Walton and Krabbe [25] call *deliberation* dialogues. These are dialogues in which two or more agents converse to formulate a joint course of action.

At the time of writing, we have team formation dialogues Dignum *et al.* [5], dialogues about what should be done [8], dialogues in which one agent proposes a plan and then persuades others to adopt it [17], and even a general purpose framework for

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 229–244, 2006.

deliberation [12]. Our goal in this paper is to provide a form of dialogue which allows agents to exchange arguments about the details of planning. This ability to debate the details is something we believe is essential if agents are going to rationally discuss what plans to adopt. In particular, we aim to develop a dialogue in which agents not only decide what to do, but create a plan *jointly*, with different sub-plans being suggested by different agents which then merge them to create an overall plan that they all agree on. This is an important step towards a complete account of deliberation.

2 Notation

As usual when considering planning, whether the classical planning of STRIPS or the decision theoretic planning of POMDPS [1], we abstract the physical world into states, actions, and state transitions caused by the actions. States and actions are the basic objects in L, the underlying language used in our approach. The kind of procedure we are interested in will determine how to compose a sequence of actions to achieve a desired state transition, namely to reach a goal from a given state.

In L, we think of a plan as being a sequence of actions, and we want to determine a plan that gets us from a specified initial state to a specified final state. The basic objects of L are:

1. A set of states: $S = \{s_0, s_1, \ldots, s_n\}$.
2. A set of actions: $A = \{a_0, a_1, \ldots, a_m\}$.
3. A set of pairs of states, where each pair consists of a start state and an end state. We term such a pair a *nisus*[1] and denote a set of nisi: $N = \{s_0 \hookrightarrow s'_0, s_1 \hookrightarrow s'_1, \ldots, s_t \hookrightarrow s'_t\}$ where $s_i \in S$ and $\hookrightarrow$ denotes a state transition. By way of an abbreviation, we sometimes write N as $N = \{n_0, n_1, ..., n_t\}$ where $n_i = s_i \hookrightarrow s'_i$.
4. A set of plans: $P = \{p_0, p_1, .., p_s\}$. A plan p is a sequence of actions $p = a_1, \ldots, a_t$ where every $a_i \in A$.

Sentences in L describe the world. In particular we are interested in what actions and plans achieve. The effect of an action can be thought of as either:

- causing a state transition; or
- achieving a nisus, denoted $a \rightsquigarrow (s \hookrightarrow s')$, where $s, s' \in S$, $a \in A$, indicating that action a achieves a transition from s to s'. This can also be written $a \rightsquigarrow n$ for nisus n.

In other words, actions bring about simple state transitions, and some of these state transitions may be distinguished as nisi — transitions between states we identify as start and end points for agents.

The effect of a plan can also be thought of in terms of state transitions: $p \rightsquigarrow (s \hookrightarrow s')$, where $p \in P$, means plan p causes a state transition from s, the *source*, to s', the *destination*. Here p must satisfy the conditions that:

- $p = a_1, \ldots, a_t$; and
- there exists a sequence of states $s_0, s_1, \ldots, s_{t-1}, s_t$ such that $s = s_0$, $s' = s_t$, and $a_i \rightsquigarrow (s_{i-1} \hookrightarrow s_i)$ for $i = 1, \ldots, t$.

[1] Nisus: a striving towards a goal.

Plans are thus specific sequences of actions that, when executed[2], will create a path through state space between two specified states. Our notion of plan is thus very much like the usual notion of a plan in simple AI planning. However, we choose to specify goals not by the usual target state but rather as a pair of initial state and final state (though we also use the conventional notion of goal in places). Why do we do this?

The answer is that the notion of a nisus fits rather better with our approach to deliberation than the usual notion of a goal. Obviously, plans, goals and nisi are all suitable for describing a state-transition graph $G = (S, A)$ where nodes S are states, edges A are assigned by atomic actions, and edges are directed[3]. Planning is essentially a process of finding a path between two given states, s_0 and s_g, and were we to plan in classic means-ends style, the usual notion of goal would suffice. However, we don't. The planning process we describe allows agents to work from initial state, goal state, or between any states in the middle, and in such a situation it is convenient to be able to link start and end point exactly as a nisus does, in order to keep track of where one is in the process.

Our slightly non-standard notation for action, $a \rightsquigarrow (s_1 \hookrightarrow s_2)$, is similarly motivated by the deliberation process. A set of action descriptions of the kind we use can be viewed as the definition of a function $a : S \rightarrow S$. By taking an action to be a function on states in this way, we capture explicitly, and in a propositional form, the fact that the same action applied to different initial states will lead to different final states (and it is a short step to capturing non-deterministic actions).

In the rest of the paper we blur the distinction between actions and plans because we take an action a to be an atomic plan — both actions and plans have the effect of creating state transitions.

3 Deliberation and Planning

We start by considering the deliberation process that a single agent goes through. The usage that Walton and Krabbe [25] make of the term "deliberation", which is to denote the whole scope of practical reasoning, differs from that made by Bratman [2], who uses it to denote the process of choosing goals[4] that are then subject to means-ends reasoning. Since we will be considering both types of deliberation, we will denote the first by D_{WK}, and the second by D_B. For us, the D_{WK} process starts with an overall nisus n_0, and uses D_B to refine the set of sub-nisi in conjunction with a process of means-ends planning.

In more detail, we recursively divide D_B and the associated planning into *phases* until a plan for n_0 is reached. All the phases share the same top-level nisus n_0, and each phase has a *deliberation context*. A deliberation context consists of:

1. the top-level nisus n_0,
2. a set of intermediate nisi N_{inter} and,
3. a set of useful plans P_{useful} (we will describe the way this set is constructed in detail below).

[2] Assuming, as we do for now, that actions are deterministic.

[3] An action can assign more than one edge between nodes.

[4] To be precise Bratman considers intentions not goals, but for our purposes there is little practical difference.

Within a context, an agent deliberates in the D_B sense. Based on the result of this D_B deliberation, the agent then plans. Based on the results of D_B and the subsequent planning, the agent then decides whether or not to recursively call a *child phase* to solve a sub-problem. If this is the case, the next round of D_B and planning is delayed until the child phase is complete.

To make the procedure precise, we define the following:

1. $\mathit{Justified}(n)$ means that a nisus n is achievable, namely it is a nisus with a plan p such that $p \leadsto n$.
2. $\mathit{Src}(N) = \{n | n \hookrightarrow n' \in N\}$ is the set of source states of a given set of nisi N.
3. $\mathit{Dest}(N) = \{s' | s \hookrightarrow s' \in N\}$ is the set of destination states of a given set of nisi N.
4. $\mathit{Src}(P) = \{s | p \leadsto (s \hookrightarrow s'), p \in P\}$ is the set of source states of a given set of plans P.
5. $\mathit{Dest}(P) = \{s' | p \leadsto (s \hookrightarrow s'), p \in P\}$ is the set of destination states of a given set of plans P.

Now, we present our first D_{WK} procedure, SD (for simple deliberation). This is called with a top-level nisus $n_0 = s_0 \hookrightarrow s_g$, initializes its top-level context with context ID $i = 0$, $N^i_{inter} = \emptyset$, and a set of partial plans that might be adopted $P^i_{useful} = A$. SD then executes the following steps:

1. Check whether $\mathit{Justified}(n_0)$ holds, that is whether n_0 can be achieved using plans in P^i_{useful}. If it can, then stop with a plan for n_0.
2. Carry out D_B:
   (a) Create a child context ID j in order to invoke a child phase.
   (b) Choose a set of intermediate nisi N^j_{inter} for the child phase from:

   Nisi of the form $s \hookrightarrow s'$ where $s \in \mathit{Dest}(P^i_{useful})$ and $s' \in \mathit{Src}(P^i_{useful})$.

   > These are all the possible state pairs which connect the end state of one existing plan to the start state of another existing plans. If there are plans for these nisi output by the planning procedure in the future, then we can create new plans by combining two existing plans with these future plans.

   Nisi of the form $s_0 \hookrightarrow s'$ where $s' \in \mathit{Src}(P^i_{useful})$.

   > These are all the possible state pairs which connect initial state of the top-level nisus to the start state of the existing plans.

   Nisi of the form $s \hookrightarrow s_g$ where $s \in \mathit{Dest}(P^i_{useful})$.

   > These are all the possible state pairs which connect the end state of the existing plans to the goal state of the top-level nisus.

   Based on the candidate nisi given above, heuristics (see below) are used to gather a subset of these nisi to be N^j_{inter}, nisi that are believed to be important to achieve our top-level nisus n_0.

   Note that the last two sets of nisi are not necessary, but help to make the process more efficient.

3. Combine plans guided by the result of D_B. For each intermediate nisus $s \hookrightarrow s' \in N^j_{inter}$, combine plans as follows:
   (a) Extend forward plans that end with initial states of nisi in N^j_{inter}: look for plans $p_1 \leadsto (s \hookrightarrow s_i)$ and $p_2 \leadsto (s_i \hookrightarrow s_j)$ and combine them to give $p_1, p_2 \leadsto (s \hookrightarrow s_j)$ if such p_1, s_i, s_j, p_2 exist.
   (b) Extend backward plans that start with final states of nisi in N^j_{inter}: look for plans $p_1 \leadsto (s_i \hookrightarrow s_j)$ and $p_2 \leadsto (s_j \hookrightarrow s')$ and combine them to give $p_1, p_2 \leadsto (s_i \hookrightarrow s')$ if such p_1, s_i, s_j, p_2 exist.

   Add these plans into P^i_{useful}.
4. Reason about plans. Pick a subset of P^i_{useful} to be the set P^j_{useful} passed to a child phase which recursively applies the SD procedure with the new context with ID j. Repeat until all the plans in P^i_{useful} have been distributed.
5. Collect plans from child phases and combine all the P^j_{useful} into P^i_{useful}.
6. Terminate if it is clear there is no plan to achieve n_0, otherwise go to the beginning of the procedure.

Step 2(b) is the key step in the planning process—each of the sub-parts of this step are ways in which the plan is constructed. The second and third sub-parts, respectively, capture the notions of backward chaining from the goal state and forward chaining from the initial state. The first step captures the idea that planning can work forwards and backwards simultaneously from some state in the middle of a possible plan.

Although we are describing this process for a single agent at the moment, consider how such a process might take place were several agents to be involved. In such a case different agents would be throwing out different suggestions simultaneously, and at any one time, we might have plans for achieving many different nisi “on the table”. The heuristics in the fourth sub-part of 2(b), are methods that select the most promising of such a set of nisi (which can equally well be identified by a single agent) for further consideration.

Below we will adapt this procedure first to incorporate argumentation, and then to allow it to be distributed across a pair of agents. Before we do this, we obtain soundness and completeness results:

Proposition 1 (Soundness). *If* SD *generates a plan p, then p is a plan to achieve the top-level nisus n_0 using the atomic actions A.*

Proof. Step 3 *of the deliberation procedure ensures that only valid plans in L are composed from actions in A. Step* 1 *guarantees that the deliberation procedure succeeds only if there is a plan $p \leadsto n_0$. Therefore p is a valid plan to achieve n_0 using the atomic actions A.* □

Before attempting the completeness result, we need some further notation:

1. P_n is the set of plans that include n actions.
2. $\oplus$ is a plan combination operator $P_i \oplus P_j = \{p_1, p_2 | p_1 \in P_i \wedge p_2 \in P_j\} \cup \{p_2, p_1 | p_1 \in P_i \wedge p_2 \in P_j\}$ where p_1, p_2 and p_2, p_1 must satisfy the valid plan conditions given above. This operator corresponds to step 3.

Proposition 2 (Completeness). *If there is a plan for initial nisus n_0, then* SD *will succeed with a plan p which achieves n_0.*

Proof. In step 2*,* P^i_{useful} *determines* N^j_{inter} *(*j *is a child context of* i*). In step* 3*,* N^j_{inter} *determines the plans being added into* P^i_{useful}*. Therefore step* 2 *and* 3 *together determine the growth of* P^i_{useful}*. The recursive child phases called in step* 5 *expedite the discovery of the top-level nisus* n_0*; they don't affect the growth of* P^i_{useful}*. We will show that step* 2 *and step* 3 *together will grow* P^0_{useful} *to contain all the plans which can be generated from atomic actions* A *so that if there is a plan for* n_0 *the deliberation procedure will certainly discover it.*

We divide P^0_{useful} *into* n *disjoint subsets* $P = P_1 \cup P_2 \cup ... \cup P_n$*, where* n *is the number of states, thus defining the maximum length of plans in* P_n[5]*. Initially* $P^0_1 = A$ *and* $P^0_i = \emptyset$ *for* $i = 2, \ldots, n$*. At each point in time, steps* 2 *and* 3 *together grow* P *in the following way:* $P^{k+1}_t = \cup(P^k_i \oplus P^k_j)$ *for all* $i + j = t$*. Therefore if, in step* 2 *and* 3*, the* P^k_i *are fixed for* $i = 1, \ldots, t-1$ *then* $P^{k+1}_t = P^k_t$*.* P_1 *is fixed during any iteration; after the first iteration* P_2 *is fixed since* P_1 *is fixed; after the second iteration* P_3 *is fixed since* P_1 *and* P_2 *are fixed, and so on. In this way, after* $n-1$ *iterations,* P_n *will be fixed. Since the maximal plan length is* n*,* P *will contain all the possible plans after* $n-1$ *iterations. Therefore if there is a valid plan for* n_0*, then* P *will contain it after* $n-1$ *iterations.* □

4 Argument and Deliberation

To combine D_B with argumentation, we need to do three things. First, we extend L with predicates that control the D_B procedure. Second, we establish logic-based rules for handling nisi, reasoning about plans, combining plans and passing information through different contexts. (This will enable us to construct plans by STRIPS-like logical reasoning). Third, we add a commitment store [11] to track the course of D_B and, hence, the course of the planning process.

With a knowledge base expanded using the extended L, a plan for a nisus is certainly contained in the theorems of a subset of the knowledge base. However, the deliberation problem, to some extent, is to select an efficient way to construct a proof which backs up a plan for a nisus (the proof then becomes the justification that can be provided in a multi-agent D_{WK}). The commitment store provides a trace of how such a proof is constructed.

4.1 Additional Notation

To capture the context of phases, we introduce the following predicates into L

1. $Ultimate(n)$ denotes that $n \in N$ is the top-level nisus.
2. $N(id, n)$ denotes that $n \in N$ is an intermediate nisus in context with ID id. It is a predicate that determines whether $n \in n^{id}_{inter}$.

[5] In any plan, we always discard any action sequence that includes a cycle $p = a_1, a_2, \ldots a_{i-1}, a_i, \ldots, a_i, a_{i+1}, \ldots$, because if we can construct the former plan, we will have sufficient plan fragments to eventually construct the corresponding contracted plan $p' = a_1, a_2, \ldots a_{i-1}, a_i, a_{i+1} \ldots$. Doing this effectively makes the Markov assumption, taking the effects of an action to uniquely determine the succeeding state.

3. $P(id, p)$ denotes that $p \in P$ is a useful plan in context with ID id. It is a predicate that determines whether $p \in P^{id}_{useful}$ or not.
4. $\mathit{Justified}(id, n)$ denotes the existence of a plan for nisus n in a the context id.
5. $\mathit{Parent}(id_1, id_2)$ denotes the fact that context id_1 is the parent of context id_2.

4.2 Rules

In order to create arguments that support plans, we need to be able to trace the planning process. To do that we need to introduce the following logical rules.

Nisus Justification

$$P(i, p) \wedge [p \rightsquigarrow (s \hookrightarrow s')] \rightarrow \mathit{Justified}(i, s \hookrightarrow s')$$

Note that here, as in all these rules, → denotes material implication.

Candidate Nisus Composition

$$\mathit{Ultimate}(n) \rightarrow N(j, n)$$

$$\begin{aligned} & \mathit{Parent}(i, j) \\ & \wedge P(i, p_1) \\ & \wedge P(i, p_2) \\ & \wedge [p_1 \rightsquigarrow (s' \hookrightarrow s_{foo1})] \\ & \wedge [p_2 \rightsquigarrow (s_{foo2} \hookrightarrow s)] \ \rightarrow N_{cand}(j, s \hookrightarrow s') \end{aligned}$$

$$\begin{aligned} & \mathit{Parent}(i, j) \\ & \wedge P(i, p) \\ & \wedge \mathit{Ultimate}(s_0 \hookrightarrow s_g) \\ & \wedge [p \rightsquigarrow (s \hookrightarrow s_{foo})] \quad \rightarrow N_{cand}(j, s_0 \hookrightarrow s) \end{aligned}$$

$$\begin{aligned} & \mathit{Parent}(i, j) \\ & \wedge P(i, p) \\ & \wedge \mathit{Ultimate}(s_0 \hookrightarrow s_g) \\ & \wedge [p \rightsquigarrow (s_{foo} \hookrightarrow s)] \quad \rightarrow N_{cand}(j, s \hookrightarrow s_g) \end{aligned}$$

We can use heuristics to select $N(j, n)$ from $N_{cand}(j, n)$ in order to reduce the search space. Without the heuristics, we will use the rule

$$N_{cand}(j, n) \rightarrow N(j, n)$$

so that every candidate nisus is considered.

Candidate Plan Combination

$$\begin{aligned} & [p_1 \rightsquigarrow (s \hookrightarrow s_m)] \\ & \wedge [p_2 \rightsquigarrow (s_m \hookrightarrow s')] \rightarrow p_1, p_2 \rightsquigarrow (s \hookrightarrow s') \end{aligned}$$

$$\begin{aligned}&P(i, p_1)\\ &\wedge P(i, p_2)\\ &\wedge N(i, s \hookrightarrow s_{foo})\\ &\wedge p_1, p_2 \rightsquigarrow (s \hookrightarrow s') \rightarrow P(i, p_1, p_2)\end{aligned}$$

$$\begin{aligned}&P(i, p_1)\\ &\wedge P(i, p_2)\\ &\wedge N(i, s_{foo} \hookrightarrow s)\\ &\wedge p_1, p_2 \rightsquigarrow (s \hookrightarrow s') \rightarrow P(i, p_1, p_2)\end{aligned}$$

Plan Selection

$$Parent(i, j) \wedge P(i, p) \rightarrow P_{cand}(j, p)$$

Again we can use heuristics select $P(j, p)$ from $P_{cand}(j, p)$. Without using heuristics, we will have rule

$$P_{cand}(j, p) \rightarrow P(j, p)$$

so that every candidate plan is considered.

Plan Collection

$$P(j, p) \wedge Parent(i, j) \rightarrow P(i, p)$$

These basic rules provide a backbone to guarantee that our procedure searches the whole space of plans so that if there is a plan to achieve the nisus n_0 then we will reach it sooner or later.

4.3 Heuristics

The basic rules give us a no-frills planning procedure. Adding in heuristics like those given below tries to ensure that if there is a plan that can achieve the top-level nisus, the deliberation procedure will reach it as early as possible. We take inspiration from decision-theoretic planning [1], where choices between actions are made on the basis of their expected cost. Accordingly we introduce the following notions of cost.

1. The *action-state transition cost* $cost(a, s, s')$ is the cost of taking action a to transform state from s to s'. The value is computed or assigned outside the reasoning system.
2. The *plan-state-transition cost* $cost(p, s, s')$ is the cost of taking a plan p to transform state from s to s'. The value is computed from $cost(a, s_i, s'_{i+1})$ for all actions a in the plan p.

The *overall cost* of a plan is computed from $cost(p, s_i, s_j)$ for all the plans p that can cause state transition from s_i to s_j. We can think of the overall cost either as $cost(s, s')$, the cost of transforming state s into s', or as $cost(i, p)$, the cost of executing plan p in context i. The idea is that although we often want to consider $p \rightsquigarrow (s \hookrightarrow s')$ as a holistic entity, to do D_B and planning we need to make comparisons between plans and actions, and we use costs to make these comparisons.

The cost of a plan can be derived from the cost of its actions in the same kind of way as it is done in decision-theoretic planning[6]. Whatever mechanism is adopted, it is outside the logical reasoning that we are studying here. Thus the assignment of costs to overall plans is, so far as the D_B and planning processes are concerned, carried out by an oracle.

Another useful notion in deciding which nisi to adopt is the *correlated valuation* of one nisus relative to another, denoted $value(n, n')$. This captures the value of achieving nisus n' in order to achieve n, and can be computed from the costs of all the plans which have the form $p \leadsto n$ for which there exists a subplan p' of p such that $p' \leadsto n'$. If there is no such sub-plan p' then $value(n, n') = 0$.

With these ideas in place, we can suggest heuristics for plan selection and nisus composition. One possibility for plan selection is to select the lowest cost plan:

$$\begin{aligned}
&Parent(i, j)\\
&\wedge P_{cand}(j, p)\\
&\wedge P(i, p)\\
&\wedge P(i, p')\\
&\wedge cost(i, p) < cost(i, p') \rightarrow P(j, p)
\end{aligned}$$

A possibility for nisus composition is to only adopt nisi for which the correlated valuation is above some threshold. To do this we can use:

$$\begin{aligned}
&N(j, n)\\
&\wedge N_{cand}(j, n')\\
&\wedge value(n, n') > c \rightarrow N(j, n')
\end{aligned}$$

Other heuristics can, of course, be adopted.

4.4 Single Agent Deliberation

We are now in a position to explain how a single agent can use argumentation-based D_{WK} to figure out what to do. We assume the agent has a knowledge base KB which contains a description of the physical world (e.g. the set of available actions and their effects). The agent also has a commitment store CS which it uses to trace the course of deliberation. The idea behind the procedure is to guarantee that all necessary sentences to support an argument are available in the commitment store CS before such an argument is constructed. The argumentation system used is

$$\mathcal{AS} = \langle \mathcal{A}(KB \cup CS), Undercut, Pref \rangle$$

in the notation of [19].

The procedure for argumentation-based deliberation, SDA (Simple Deliberation through Argumentation), is given a top-level nisus $n_0 = s_0 \hookrightarrow s_g$. It first initializes the context id with $i = 0$, CS with $Ultimate(n_0)$ and $P(i, a)$ for all $a \in A$, then executes the following steps:

[6] Such costs may be assigned by a form of reinforcement learning, for example.

1. Check $\mathcal{AS}$ to see whether $\mathit{Justified}(n_0)$ is acceptable. If it is, then stop with a plan for n_0 in CS.
2. Carry out a D_B:
   (a) Set a context ID j for a child phase.
   (b) Using KB and CS, use $\mathcal{AS}$ to check if $N(j, n)$ is acceptable for nisi of the following three kinds:

   Nisi such as $(s \hookrightarrow s')$ for $s \in \mathit{Dest}(P)$ and $s' \in \mathit{Src}(P)$. This captures the idea of extending existing plans forwards and backwards.

   Nisi such as $(s_0 \hookrightarrow s')$ for $s' \in \mathit{Src}(P)$. This captures the idea of extending the existing plans forwards from the source of the top-level nisus.

   Nisi such as $(s \hookrightarrow s_g)$ for $s \in \mathit{Dest}(P)$ to capture the idea of extending the existing plans backwards from the destination of the top-level nisus.

   Assert all the acceptable sentences $N(j, n)$ into CS. Notice that the above can be achieved only if the rules for nisus composition are used. If the agent exhausts all the possible candidates nisi $N_{cand}(j, n)$ but no $G(j, n)$ can be asserted, then the child phase returns to the parent context with no new plans added.
3. Combine plans guided by the results of D_B. For all the nisi $n = (s \hookrightarrow s')$ with acceptable arguments for $N(j, n)$, combine plans as follows:

   Plans from s. Look for plans $p_1 \rightsquigarrow (s \hookrightarrow s_i)$ and $p_2 \rightsquigarrow (s_i \hookrightarrow s_j)$, combine them to be $p_1, p_2 \rightsquigarrow (s \hookrightarrow s_j)$ if such p_1, s_i, s_j, p_2 exist.

   Plans to s'. Look for plans $p_1 \rightsquigarrow (s_i \hookrightarrow s_j)$ and $p_2 \rightsquigarrow (s_j \hookrightarrow s')$, combine them to be $p_1, p_2 \rightsquigarrow (s_i \hookrightarrow s')$ if such p_1, s_i, s_j, p_2 exist.
   Check acceptability of plans with $\mathcal{AS}$, and assert all acceptable $p_1, p_2 \rightsquigarrow (s \hookrightarrow s_j)$, $p_1, p_2 \rightsquigarrow (s_i \hookrightarrow s')$ into CS.
4. Reason about plans. Using the plan selection rules, identify the candidate useful plans p, and for each use $\mathcal{AS}$ to check whether $P(j, p)$ is acceptable. If it is acceptable, assert it into CS.
5. Recursively call a child phase to go through SDA with the new context with ID j.
6. Collect plans from the child phase: Use plan collection rules to identify candidates plans p. For each p, check with $\mathcal{AS}$ whether $P(i, p)$ is acceptable. If it is acceptable, assert it into CS.
7. Go to the first step.

Since SDA is based on a process for D_B and planning that we know is sound and complete, we can easily show that it is sound and complete itself:

Proposition 3 (Soundness). *If plan p for nisus n_0 is acceptable according to AS at the end of the* SDA*, then p achieves n_0 using actions from A.*

Proof. *Step 3 combines plan p from P_1 and P_2 only if AS accepts the combination. Thus AS accepts the effects of p. Step 1 ensures that p is a plan for n_0.* □

Proposition 4 (Completeness). *If there is a plan p which achieves nisus n_0 using atomic actions A, then* SDA *will generate p.*

Proof. *Similar to the proof of Proposition* 2*, since steps* 2 *and* 3 *together will explore all the possible combinations of discovered acceptable plans, the procedure will generate all the plans acceptable by* $\mathcal{AS}$ *at the end of the procedure. If there is plan* p *which achieves initial nisus* n_0 *using actions from* A*, then such a plan is contained in all the acceptable plans by* $\mathcal{AS}$*.* □

5 Deliberation Dialogues

We now consider how to extend the D_{WK} process to become a dialogue. We describe the dialogue process as being between just two agents, but it can easily be extended to a multi-party dialogue.

5.1 Basic Configuration

The scenario for which our deliberation dialogue was created is as follows:

1. Dialogues take place between two agents, $A1$ and $A2$.
2. $A1$ has initial knowledge base KB_1 and commitment store CS_1.
3. $A2$ has initial knowledge base: KB_2 and a commitment store CS_2.
4. $A1$ and $A2$ share the same rules for planning and deliberation but differ in the way they evaluate plans and nisi. They may also have different set of actions reflecting different capabilities.
5. The context ID, id, is shared by $A1$ and $A2$. Initially, $id = 0$.
6. $A1$ and $A2$ have an mechanism to allocate unique context IDs.
7. Both $A1$ and $A2$ can access CS_1 and CS_2, hence the argumentation system of $A1$ is
$$\mathcal{AS}_1 = \langle \mathcal{A}(KB_1 \cup CS_1 \cup CS_2), \mathit{Undercut}, \mathit{Pref} \rangle$$
and the argumentation system of $A2$ is
$$\mathcal{AS}_2 = \langle \mathcal{A}(KB_2 \cup CS_1 \cup CS_2), \mathit{Undercut}, \mathit{Pref} \rangle.$$

Within this scenario we need to add the following idea of an auxiliary sub-dialogue.

5.2 Auxiliary Discussion Sub-dialogue

One of the reasons for agents to engage in deliberation dialogues is to combine both agents' reasoning and planning capabilities. One possible downside, the fact that conflicts may arise between the two agents, can be resolved by the use of argumentation (which, at heart, is a system for resolving conflicts in terms of the acceptability of the arguments that support the conflicting statements in two argumentation systems). To achieve this resolution we need an auxiliary discussion sub-dialogue to render a sentence acceptable to both agents, by which we mean that it is accepted by the argumentation systems $\mathcal{AS}_1$ and $\mathcal{AS}_2$.

A discussion sub-dialogue is started by a dialogue move $\mathit{discuss}(p)$. Assuming that $A1$ moves first, the discussion sub-dialogue proceeds as follows:

1. $A1$ checks with its own argumentation system AS_1 whether p is acceptable. If it is, then $A1$ makes the locution $discuss(p)$, indicating that p is open for discussion.
2. $A2$ checks with its argumentation system AS_2 whether p is acceptable. If it is, then $A2$ stops and declares that p is accepted by both agents. Otherwise, $A2$ challenges p, indicating that it needs to see the argument for p (which will be the reason behind A_1's suggestion of p).
3. $A1$ responds to the challenge by asserting the set of support S for p.
4. For each sentence $q' \in S$, $A2$ checks with AS_2. For the unaccepted sentences $q' \in S$, $A2$ discusses $\neg q'$ with $A1$, if any of the $\neg q'$ are accepted by the discussion then $A1$ goes to next step; otherwise $A2$ stops and declares p is accepted by both agents.
5. $A1$ replaces S with another alternative support and goes back to step 3.
6. If all the possible supports for p are put forward by $A1$, but the discussion with $A2$ accepts none of them, then $A1$ declares that p is not accepted by both agents.

With this machinery, we can now set down the deliberation dialogue.

5.3 A Dialogue for Deliberation

Two agents can have two different set of atomic actions. This means they have different capability or different views of the physical world.

We assume that initially $A1$ and $A2$ agree on a top-level nisus $n_0 = s_0 \hookrightarrow s_g$ (though they could arrive at this after another dialogue about what they want, a negotiation perhaps). $A1$ and $A2$ initialize the context id with $i = 0$. $A1$ initializes CS_1 with $\mathit{Ultimate}(n_0)$ and $P(i, a)$ for all its actions, and $A2$ does the same for its commitment store. The simple deliberation dialogue, SDD, then consists of the following steps:

1. Check a plan is required.
   $A1$ discusses $A2$ to check whether $\mathit{Justified}(n_0)$ is acceptable. If it is, then the agents stop with a plan for n_0. Otherwise proceed to the next step.
2. Carry out D_B:
   (a) $A1$ and $A2$ create a context ID j for a child phase.
   (b) $A1$ discusses with $A2$ to check whether any nisi $N(j, n)$ of the following three forms are acceptable:

   Nisi of the form $s \hookrightarrow s'$ where $s \in \mathit{Dest}(P)$ and $s' \in \mathit{Src}(P)$, to capture the idea of extending existing plans forwards and backwards.

   Nisi of the form $s_0 \hookrightarrow s'$ where $s' \in \mathit{Src}(P)$, to capture the idea of extending the existing plans forwards from the source of the top-level nisus.

   Nisi of the form $s \hookrightarrow s_g$ where $s \in \mathit{Dest}(P)$, to capture the idea of extending the existing plans backwards from the destination of the top-level nisus.

   $A1$ asserts all the acceptable sentences $N(j, n)$ into CS_1.
   (c) $A2$ carries out step (b) but asserts results into CS_2. Completeness hinges on doing this — otherwise some crucial information known only to $A2$ might not be available (to $A1$ which can use it to construct the plan).

After $A1$ and $A2$ have exhausted all the nisi n that satisfy $N_{cand}(j, n)$, and no new $N(j, n)$ are asserted, they return to the parent context.

3. Combine plans guided by the results of D_B.
   (a) For all the nisi $n = (s \hookrightarrow s')$ with acceptable arguments for $N(j, n)$, $A1$ combines plans:

   Plans from s: $A1$ looks for plans $p_1 \rightsquigarrow (s \hookrightarrow s_i)$ and $p_2 \rightsquigarrow (s_i \hookrightarrow s_j)$, and combines them to form $p_1, p_2 \rightsquigarrow (s \hookrightarrow s_j)$ if such p_1, s_i, s_j, p_2 exist.

   Plans to s': $A2$ looks for plans $p_1 \rightsquigarrow (s_i \hookrightarrow s_j)$ and $p_2 \rightsquigarrow (s_j \hookrightarrow s')$, and combines them to be $p_1, p_2 \rightsquigarrow (s_i \hookrightarrow s')$ if such p_1, s_i, s_j, p_2 exist.

   $A1$ discusses $p_1, p_2 \rightsquigarrow (s \hookrightarrow s_j)$, $p_1, p_2 \rightsquigarrow (s_i \hookrightarrow s')$, with $A2$, asserting the acceptable plans into CS_1.

   (b) $A2$ does the same as $A1$ does for (a) but asserts the results into CS_2.
4. Reason about plans:
   (a) $A1$ uses the plan selection rules to identify candidate useful plans p, and for such plans discusses with $A2$ whether $P(j, p)$ is acceptable. If one or more p are acceptable, $A1$ asserts them into CS_1.
   (b) $A2$ carries out the analogous process.
5. Recursively call child phases to go through SDD with a new context.
6. Collect plans from the child phases:
   (a) $A1$ uses the plan collection rules to figure out which candidate plans p should be collected. Then it discusses with $A2$ to check whether the resulting $P(i, p)$ are acceptable. If one or more are acceptable, then they are asserted into CS_1.
   (b) $A2$ carries out the analogous process
7. Go to the beginning of the procedure.

Once again we can prove the soundness and completeness of the procedure, showing that recasting it as a dialogue does not detract from it:

Proposition 5 (Soundness). *If plan p for nisus n_0 is acceptable by both agents at the end of* SDD*, then p achieves n_0 using atomic actions that both agents agree upon.*

Proof. Step 3 *combines plan p from P_1 and P_2 only if $\mathcal{AS}_1$ and $\mathcal{AS}_2$ both accept the combination. Thus both agents agree on the effects of p. Step* 1 *ensures that both agents agree that p is a plan for n_0.* □

Proposition 6 (Completeness). *If there is a plan p which achieves nisus n_0 using a set of atomic actions that both agents agree upon, then* SDD *will generate p.*

Proof. Similar to the proof of Propositions 2 and 4, since steps 2 and 3 together will explore all the possible combinations of discovered plans accepted by both agents, the procedure will generate all the plans acceptable by both $\mathcal{AS}_1$ and $\mathcal{AS}_2$ at the end of the procedure. If, according to the acceptable atomic actions A agreed by both agents, there is plan p which achieves initial nisus n_0, then such a plan is contained in all the acceptable plans by $\mathcal{AS}_1$ and $\mathcal{AS}_2$. □

6 Discussion

The full argumentation-based deliberation dialogue SDD successfully composes plans in one of the following ways:

1. $A1$ composes the whole plan, $A2$ agrees with it.
2. $A2$ composes the whole plan, $A1$ agrees with it.
3. $A1$ composes some parts of the plan; $A2$ composes some other parts of the plan; $A1$ and $A2$ combine the two parts to create a partial plan that both $A1$ and $A2$ agree on, and so on until the whole plan is constructed.

Thus we can see that our process combines the "create a plan and then convince others it works" approach of [17] with the "merge different plans" approach of [7]. For a given situation, SDD will typically take an approach that is a mixture of the two, involving the creation and merging of separate sub-plans, some of which may be as simple as a single action. Some of this merging will involve one agent persuading the other to adopt the sub-plan. SDD thus achieves the overall goal that we set out at the start of the paper.

One thing to note about this work concerns the heuristics used to guide the search for plans during deliberation. We never specified these in any detail, though we give some high-level hints about the possible form that they may take. This does not detract from the formal results, since the results hold even if we have no heuristics (in which case we essentially do an exhaustive search through the full state-space). However, decent heuristics will help to focus the search and thus make it more efficient.

Finally, we should note that there are clearly some similarities between the way in which our approach to deliberation allows plans to grow around any nisus (that is to grow forwards, backwards, or in both directions from somewhere in what turns out to be the middle of the plan) and partial order planning. This relationship is something we will explore in the future. It is particularly thought-provoking since the part of this work that we believe is most valuable, that is the integration of planning and multi-agent argumentation-based dialogue, does not depend upon the precise kind of planning used (we simply picked one form of planning to make our suggestion concrete). As a result, partial order planning could easily be fitted into our framework, and it would be interesting to investigate the kind of system that resulted from doing this.

7 Conclusion

This paper has described a mechanism for carrying out deliberation dialogues in the sense of Walton and Krabbe [25] at a high level of detail — that is dialogues in which agents decide what actions to perform and the order in which they are performed. Our approach, which as described is limited to two agents but could easily be generalized, recursively mixes nisus selection and planning, allowing these tasks to be distributed between the agents in a flexible way. The approach makes it possible for agents to combine their knowledge about the environment, and to make use of the planning abilities of both agents (since one can readily imagine that they have complementary expertise, as embodied in the heuristics they can employ).

Two directions of future research are particularly attractive to us. First, as mentioned above, it seems appropriate to allow the agents to learn the values of actions across a number of trials, and this might easily be achieved by techniques from reinforcement learning. Doing so suggests a bridge between the kind of procedure we have developed here and multi-agent decision theoretic planning of the kind considered in [10]. Exploring such connections is the second direction we intend to take.

Acknowledgements. This work was made possible by funding from NSF #REC-02-19347, NSF #IIS 0329037 and EU FP6-IST 002307 (ASPIC).

References

1. Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.
2. M. E. Bratman. *Intention, plans, and practical reason*. CSLI Publications, 1999.
3. D. D. Corkill. Hierarchical planning in a distributed environment. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 168–175, 1979.
4. M. E. desJardins, E. H. Durfee, C. L. Ortiz, and M. J. Wolverton. A survey of research in distributed, continual planning. *AI Magazine*, 21(1), 2000.
5. F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In C. Castelfranchi and Y. Lespérance, editors, *Seventh Workshop on Agent Theories, Architectures, and Languages*, pages 141–156, Boston, USA, 2000.
6. E. H. Durfee and V. R. Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, KDE-1:63–83, 1991.
7. E. Ephrati and J. S. Rosenschein. Multi-agent planning as the process of merging distributed sub-plans. In *Proceedings of the Twelfth International Workshop on Distributed Artificial Intelligence*, pages 115–129, 1993.
8. K. Greenwood, T. Bench-Capon, and P. McBurney. Structuring dialogue between the People and their representatives. In R. Traunmüller, editor, *Electronic Government: Proceedings of the Second International Conference (EGOV03), Prague, Czech Republic*, Lecture Notes in Computer Science 2739, pages 55–62, Berlin, Germany, 2003. Springer.
9. B. J. Grosz and S. Kraus. The evolution of SharedPlans. In M. J. Wooldridge and A. Rao, editors, *Foundations of Rational Agency*, volume 14 of *Applied Logic*. Kluwer, The Netherlands, 1999.
10. C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*, pages 1523–1530, 2001.
11. C. L. Hamblin. *Fallacies*. Methuen and Co Ltd, London, UK, 1970.
12. D. Hitchcock, P. McBurney, and S. Parsons. The eightfold way of deliberation dialogues. *International Journal of Intelligent Systems*, 2004.
13. S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.
14. H. Levesque, P. Cohen, and J. Nunes. On acting together. In *Proceedings of the National Conference on Artificial Intelligence*, pages 94–99, 1990.
15. N. Maudet and F. Evrard. A generic framework for dialogue game implementation. In *Proceedings of the 2nd Workshop on Formal Semantics and Pragmatics of Dialogue*, University of Twente, The Netherlands, May 1998.
16. P. McBurney. *Rational Interaction*. PhD thesis, University of Liverpool, 2002.
17. P. Panzarasa, N. R. Jennings, and T. J. Norman. Formalising collaborative decision making and practical reasoning in multi-agent systems. *Journal of Logic and Computation*, 12(1):55–117, 2002.

18. S. Parsons and P. McBurney. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation*, 12(5):415–439, 2003.
19. S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *1st International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 2002.
20. C. Reed. Dialogue frames in agent communications. In Y. Demazeau, editor, *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 246–253. IEEE Press, 1998.
21. M. Schroeder, D. A. Plewe, and A. Raab. Ultima ratio: should Hamlet kill Claudius. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 467–468, 1998.
22. K. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the Eleventh Joint Conference on Artificial Intelligence*, pages 517–523, 1989.
23. M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.
24. W. van der Hoek and M. Wooldridge. Tractable multiagent planning for epistemic goals. In *Proceedings of the 1st International Conference on Autonomous Agents and Multiagent Systems*, 2002.
25. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, USA, 1995.
26. R. Zlot and A. Stentz. Market-based multirobot coordination using task abstraction. In *Proceedings of the 4th International Conference on Field and Service Robotics*, 2003.

Presentation of Arguments and Counterarguments for Tentative Scientific Knowledge

Anthony Hunter

Department of Computer Science
University College London
Gower Street, London, WC1E 6BT, UK

Abstract. A key goal for a scientist is to find evidence to argue for or against universal statements (in effect first-order formulae) about the world. Building logic-based tools to support this activity could be potentially very useful for scientists to analyse new scientific findings using experimental results and established scientific knowledge. In effect, these logical tools would help scientists to present arguments and counterarguments for tentative scientific knowledge, and to share and discuss these with other scientists. To address this, in this paper, we explain how tentative and established scientific knowledge can be represented in logic, we show how first-order argumentation can be used for analysing scientific knowledge, and we extend our framework for evaluating the degree of conflict arising in scientific knowledge. We also discuss the applicability of recent developments in optimizing the impact and believability of arguments for the intended audience.

1 Introduction

Argumentation is a vital aspect of intelligent behaviour by humans. There are a number of proposals for logic-based formalisations of argumentation (for reviews see [14, 7]). These proposals allow for the representation of arguments for and against some claim, and of attack or undercut relationships between arguments. Whilst many proposals are essentially propositional, there are argumentation formalisms for reasoning with full first-order classical logic [3].

In many professional domains, such as science, it is apparent that there is a need to support first-order argumentation. For example, one of the key goals of scientists is to find evidence to argue for/against universal statements (in effect first-order formulae) about the world. Scientists have much knowledge about their area of expertise, and they have new findings which they want to consider with respect to the established knowledge. With this "knowledgebase", a scientist will often identity arguments and counterarguments for new proposals for scientific knowledge (tentative scientific knowledge). This presentation of arguments and counterarguments will be for their own analytical purposes, and for other scientists to consider and to counter.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 245–263, 2006.

Arguments and counterarguments can be systematically, though not necessarily exhaustively, identified by hand in the free text of individual scientific papers using anotation methodologies [15]. Tools have also been developed to support scientists in analysing free text arguments obtained from a collection of papers, allowing the scientist to flag relationships between evidence from different papers such as "supports", "contradicts", etc., using a graphical notation (see for example ClaimMaker [6]).

However, logic-based argumentation has not been adequately harnessed for capturing arguments and counterarguments from scientific knowledge. Potential advantages would include a more a precise representation of scientific knowledge that is tolerant of conflicts that inevitably arise, and automated reasoning for incorporation in tools for checking or generating arguments and counterarguments from scientific knowledge.

To address this need, we present a new framework for first-order argumentation with scientific knowledge. However, we are not intending to consider scientific theory formation here. Whilst argumentation theory is being considered for the process of generating new scientific theories [13], we assume that the scientist has generated a theory, and wants to analyse it with the respect to the rest of the relevant scientific knowledge.

In the following, we explain how tentative and established scientific knowledge can be represented in logic, we review our framework for first-order argumentation, we show how first-order argumentation can be used for analysing scientific knowledge, and we extend our framework for evaluating the degree of conflict arising in scientific knowledge.

2 Scientific Knowledge in Logic

Much established scientific knowledge can be represented by statements in first-order logic such as the following universal statements concerning cell biology.

$$\forall \mathtt{x}.(\mathtt{cell}(\mathtt{x}) \rightarrow \mathtt{contains}(\mathtt{x}, \mathtt{chromosomes}))$$
$$\forall \mathtt{x}.(\mathtt{chromosomes}(\mathtt{x}) \rightarrow \mathtt{contains}(\mathtt{x}, \mathtt{dna}))$$

Here we assume much established scientific knowledge derived from experimental research is represented by a set of formulae each of which is a scientific statement as defined below.

Definition 1. *A* **scientific statement** *is a closed formula of first-order logic of the following format where (1) for* $0 \leq i \leq m$*,* μ_i *is either a* $\forall$ *or a* $\exists$ *quantifier and* x_i *is a variable; and (2)* α *and* β *are conjunctions of literals.*

$$\mu_0 x_0, ..., \mu_m x_m.(\beta \rightarrow \alpha)$$

So the formulae concerning cell biology are examples of scientific statements. This is a simplistic format for scientific knowledge, but it is useful for capturing a wide range of generalities obtained from experiments or clinical drug trials, and will serve us for developing the role of logic-based argumentation in science.

A key issue in science is that established scientific knowledge is not without inconsistencies. There are competing theories and interpretations in established scientific knowledge. Furthermore, these conflicts lead to further research and hence new discoveries causing the established scientific knowledge to change over time. This is particularly so in biomedical sciences where even the more established knowledge evolves dramatically with much refinement and some established knowledge being rejected after a relatively short time period. This is manifested by the rate at which new editions of substantially revised standard undergraduate textbooks in biomedical sciences are published. It can also be seen in the rapidly evolving practices in healthcare. Some established practices are rejected in the space of a few years in the light of newly established scientific knowledge. As a result, the process of science routinely involves dealing with uncertain and conflicting information.

Scientists who consider their own experimental results in the context of the established scientific knowledge, as reflected in the scientific literature, need to reason with the conflicts arising, and determine the net results that they should put forward into the public domain, hopefully to become established scientific knowledge. But before scientific knowledge can be regarded as established, it is treated with much caution. We therefore regard findings from research as conditional knowledge, called scientific proposals, of the following form.

Definition 2. *A* **scientific proposal** *is a closed formula of first-order logic of the following format where (1) for* $0 \leq i \leq n$*,* μ_i *is either a* $\forall$ *or a* $\exists$ *quantifier and* x_i *is a variable; (2)* γ *is a conjunction of literals; and (3)* $\mu_0 x_0, ..., \mu_n x_n.(\beta \rightarrow \alpha)$ *is a scientific statement.*

$$\mu_0 x_0, ..., \mu_n x_n.(\gamma \rightarrow (\beta \rightarrow \alpha))$$

We call γ *the* **meta-condition** *and* $\mu_0 x_0, ..., \mu_n x_n.\beta \rightarrow \alpha$ *the* **tentative scientific statement** *for the scientific proposal. If* f_s *is a scientific statement, then* $\mathsf{Metacondition}(f_s) = \gamma$ *and* $\mathsf{Proposal}(f_s) = \mu_0 x_0, ..., \mu_n x_n.(\beta \rightarrow \alpha)$*.*

Whilst we do not impose any typing on the language for scientific proposals, it should be clear in the following that we intend meta-conditions to use literals that are not available for scientific statements. In general, we see a number of dimensions that we would want to define qualification (meta-conditions) for a scientific proposal. We briefly consider some examples: (1) the investigators who made the scientific contribution need to have the right qualificiations and experience; (2) the methods used in the experiments and the interpretation of the experiments need to be appropriate; and (3) the experimental results from which the tentative contribution is based do justify the tentative contribution.

We assume scientific knowledge is represented by a set of formulae of classical logic and that includes scientific statements, scientific proposals, together with subsidiary information such as details on particular experiments and particular techniques. Later we will define an argument as a minimal set of formulae (called the support) that classically implies a formula (called the consequent).

Example 1. The formula below, denoted f_1, is a scientific proposal concerning drug trial "trial78" on drug "p237" for "reducing blood cholesterol" .

$$f_1\ \forall\mathtt{x}.(\mathtt{validDrugTrial(trial78)} \rightarrow \\ (\mathtt{healthy(x)} \land \mathtt{under75(x)} \land \mathtt{treatment(x,p237,50mg,daily)} \\ \rightarrow \mathtt{decreaseBloodCholesterol(x)}))$$

The formulae f_2 and f_3 are subsidiary formulae.

$$f_2\ \forall\mathtt{x},\mathtt{y}.((\mathtt{numberOfPatients(x,y)} \land \mathtt{y} > \mathtt{1000} \land \mathtt{trialAtGoodHospital(x)}) \\ \rightarrow \mathtt{validDrugTrial(x)})$$

$$f_3\ \mathtt{numberOfPatients(trial78,2479)} \land \mathtt{2479} > \mathtt{1000} \\ \land \mathtt{trialAtGoodHospital(trial78)}$$

Assuming $\{f_1, f_2, f_3\}$ we obtain f_4 by implication.

$$f_4\ \forall\mathtt{x}.(\mathtt{healthy(x)} \land \mathtt{under75(x)} \land \mathtt{treatment(x,p237,50mg,daily)} \\ \rightarrow \mathtt{decreaseBloodCholesterol(x)})$$

This can be summarized by the following argument, where $\{f_1, f_2, f_3\}$ is the support for the argument, and f_4 is the consequent.

$$\langle\{f_1, f_2, f_3\}, f_4\rangle$$

We now turn to the kinds of counterarguments for arguments. We shall focus on undercuts. An undercut A_j for an argument A_i is an argument with a consequent that negates the support for A_i. By recursion, undercuts may be subject to undercuts. We formalize this in the next section, and then provide a framework for scientific argumentation.

3 First-Order Argumentation

In this section, we review a recent proposal for argumentation with first-order classical logic [3]. For a language, the set of formulae $\mathcal{L}$ that can be formed is given by the usual inductive definitions for classical logic. Deduction in classical propositional logic is denoted by the symbol $\vdash$ and deductive closure by Cn so that $\mathsf{Cn}(\Phi) = \{\alpha \mid \Phi \vdash \alpha\}$.

For the following definitions, we first assume a knowledgebase Δ (a finite set of formulae) and use this Δ throughout. We further assume that every subset of Δ is given an enumeration $\langle\alpha_1, \ldots, \alpha_n\rangle$ of its elements, which we call its canonical enumeration. This really is not a demanding constraint: In particular, the constraint is satisfied whenever we impose an arbitrary total ordering over Δ. Importantly, the order has no meaning and is not meant to represent any respective importance of formulae in Δ. It is only a convenient way to indicate the order in which we assume the formulae in any subset of Δ are conjoined to make a formula logically equivalent to that subset.

The paradigm for the approach is a large repository of information, represented by Δ, from which arguments can be constructed for and against arbitrary claims. Apart from information being understood as declarative statements, there is no a priori restriction on the contents, and the pieces of information in the repository can be as complex as possible. Therefore, Δ is not expected to be consistent. It need even not be the case that every single formula in Δ is consistent.

The framework adopts a very common intuitive notion of an argument. Essentially, an argument is a set of relevant formulae that can be used to classically prove some claim, together with that claim. Each claim is represented by a formula.

Definition 3. *An* **argument** *is a pair* $\langle \Phi, \alpha \rangle$ *such that: (1)* $\Phi \nvdash \bot$*; (2)* $\Phi \vdash \alpha$*; and (3) there is no* $\Phi' \subset \Phi$ *such that* $\Phi' \vdash \alpha$*. We say that* $\langle \Phi, \alpha \rangle$ *is an argument for* α*. We call* α *the consequent of the argument and* Φ *the support of the argument (we also say that* Φ *is a support for* α*). For an argument* $\langle \Phi, \alpha \rangle$*,* $\mathsf{Support}(\langle \Phi, \alpha \rangle) = \Phi$*, and* $\mathsf{Consequent}(\langle \Phi, \alpha \rangle) = \alpha$*.*

Example 2. For $\Delta = \{\forall x.(p(x) \rightarrow q(x)), p(a), \neg\forall x.p(x), \neg\exists x.(p(x) \rightarrow q(x))\}$ some arguments include

$$\langle \{p(a), \forall x.(p(x) \rightarrow q(x))\}, q(a) \rangle$$
$$\langle \{\neg\forall x.p(x)\}, \neg\forall x.p(x) \rangle$$
$$\langle \{\neg\exists x.(p(x) \rightarrow q(x))\}, \forall x.(p(x) \wedge \neg q(x)) \rangle$$

Arguments are not independent. In a sense, some encompass others (possibly up to some form of equivalence). To clarify this requires a few definitions as follows.

Definition 4. *An argument* $\langle \Phi, \alpha \rangle$ *is* **more conservative** *than an argument* $\langle \Psi, \beta \rangle$ *iff* $\Phi \subseteq \Psi$ *and* $\beta \vdash \alpha$*.*

Example 3. $\langle \{p(a), \forall x.(p(x) \rightarrow q(x) \vee r(x))\}, q(a) \vee r(a) \rangle$ is more conservative than $\langle \{p(a), \forall x.(p(x) \rightarrow q(x) \vee r(x)), \neg\exists x.r(x)\}, q(a) \rangle$.

Some arguments directly oppose the support of others, which amounts to the notion of an undercut.

Definition 5. *An* **undercut** *for an argument* $\langle \Phi, \alpha \rangle$ *is an argument* $\langle \Psi, \neg(\phi_1 \wedge \ldots \wedge \phi_n) \rangle$ *where* $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$*.*

Example 4

$$\langle \{\forall x.p(x)\}, p(a) \rangle \text{ is undercut by } \langle \{\neg\exists x.p(x)\}, \neg\forall x.p(x) \rangle$$
$$\langle \{\forall x.p(x)\}, p(a) \rangle \text{ is undercut by } \langle \{\exists x.\neg p(x)\}, \neg\forall x.p(x) \rangle$$
$$\langle \{\forall x.p(x)\}, p(a) \rangle \text{ is undercut by } \langle \{\neg p(a)\}, \neg\forall x.p(x) \rangle$$
$$\langle \{\forall x.p(x)\}, p(a) \rangle \text{ is undercut by } \langle \{\neg p(b)\}, \neg\forall x.p(x) \rangle$$

Example 5. Let $\Delta = \{p(a), p(a) \rightarrow q(a), r(a), r(a) \rightarrow \neg p(a)\}$. Then,

$$\langle \{r(a), r(a) \rightarrow \neg p(a)\}, \neg(p(a) \wedge (p(a) \rightarrow q(a))) \rangle$$

is an undercut for

$$\langle \{p(a), p(a) \rightarrow q(a)\}, q(a)\rangle$$

A less conservative undercut for it is

$$\langle \{r(a), r(a) \rightarrow \neg p(a)\}, \neg p(a)\rangle$$

Definition 6. *$\langle \Psi, \beta\rangle$ is a* **maximally conservative undercut** *of $\langle \Phi, \alpha\rangle$ iff $\langle \Psi, \beta\rangle$ is an undercut of $\langle \Phi, \alpha\rangle$ such that no undercuts of $\langle \Phi, \alpha\rangle$ are strictly more conservative than $\langle \Psi, \beta\rangle$ (that is, for all undercuts $\langle \Psi', \beta'\rangle$ of $\langle \Phi, \alpha\rangle$, if $\Psi' \subseteq \Psi$ and $\beta \vdash \beta'$ then $\Psi \subseteq \Psi'$ and $\beta' \vdash \beta$).*

The value of the following definition of canonical undercut is that we only need to take the canonical undercuts into account. This means we can justifiably ignore the potentially very large number of non-canonical undercuts.

Definition 7. *An argument $\langle \Psi, \neg(\phi_1 \wedge \ldots \wedge \phi_n)\rangle$ is a* **canonical undercut** *for $\langle \Phi, \alpha\rangle$ iff it is a maximally conservative undercut for $\langle \Phi, \alpha\rangle$ and $\langle \phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration of Φ.*

Proposition 1. *Given two different canonical undercuts for the same argument, none is more conservative than the other.*

Proposition 2. *Any two different canonical undercuts for the same argument have distinct supports whereas they do have the same consequent.*

An argument tree describes the various ways an argument can be challenged, as well as how the counterarguments to the initial argument can themselves be challenged, and so on recursively.

Definition 8. *An* **argument tree** *for α is a tree where the nodes are arguments such that*

1. *The root is an argument for α.*
2. *For no node $\langle \Phi, \beta\rangle$ with ancestor nodes $\langle \Phi_1, \beta_1\rangle, \ldots, \langle \Phi_n, \beta_n\rangle$ is Φ a subset of $\Phi_1 \cup \cdots \cup \Phi_n$.*
3. *Each child of a node A is an undercut for A that obeys 2.*

A **canonical argument tree** *is an argument tree where each undercut is a canonical undercut. A* **complete argument tree** *is a canonical argument tree for each node A, s.t. if A' is a canonical undercut for A, then A' is a child of A. For a tree T, $\mathsf{Nodes}(T)$ is the set of nodes in T and $\mathsf{Depth}(T)$ is the number of arcs on the longest branch of T.*

The second condition in Definition 8 ensures that each argument on a branch has to introduce at least one formula in its support that has not already been used by ancestor arguments. As a notational convenience, in examples of argument trees, the $\diamond$ symbol is used to denote the consequent of an argument when that argument is a canonical undercut.

Example 6. Consider the following knowledgebase.

$$\Delta = \{\forall x.(p(x) \vee q(x)), \forall x.(p(x) \rightarrow r(x)), \forall x.\neg r(x), \forall x.\neg q(x), \forall x.(s(x) \leftrightarrow q(x))$$

Below is an argument tree from Δ for the consequent $\forall x.(p(x) \vee \neg s(x))$.

$$\begin{tikzcd}
\langle\{\forall x.(p(x) \vee q(x)), \forall x.\neg q(x)\}, \forall x.(p(x) \vee \neg s(x))\rangle \\
\langle\{\forall x.(p(x) \rightarrow r(x)), \forall x.\neg r(x)\}, \diamond\rangle \arrow[u]
\end{tikzcd}$$

Example 7. Let f_5 and f_6 be the following formulae.

$$f_5\ \forall \mathtt{x}, \mathtt{y}.(\mathtt{irregularitiesDuringTrial(X)} \\ \rightarrow \neg\mathtt{validDrugTrial(x)})$$

$$f_6\ \mathtt{irregularitiesDuringTrial(trial78)}$$

Hence we have the argument $\langle\{f_5, f_6\}, \diamond\rangle$ which is an undercut for the argument $\langle\{f_1, f_2, f_3\}, f_4\rangle$ given in Example 1. This is summarized as follows.

$$\begin{tikzcd}
\langle\{f_1, f_2, f_3\}, f_4\rangle \\
\langle\{f_5, f_6\}, \diamond\rangle \arrow[u]
\end{tikzcd}$$

A complete argument tree is an efficient representation of all the important arguments and counterarguments.

Proposition 3. *Let $\alpha \in \mathcal{L}$. If Δ is finite, there is a finite number of argument trees with the root being an argument with consequent α that can be formed from Δ, and each of these trees has finite branching and a finite depth.*

4 Scientific Argumentation

After delineating some conflicting (i.e. inconsistent) scientific knowledge, we assume a scientist wants to see if an argument of interest has undercuts, and by recursion, undercuts to undercuts. So when a scientist considers a scientific proposal, undercuts to an argument using the scientific proposal indicate reasons to doubt the proposal, and undercuts to an undercut indicate reasons to doubt that undercut. Argument trees therefore provide a systematic means for representing caution in scientific knowledge. We focus on three types of undercut that arise with a clear aetiology.

One way to reflect caution in a scientific proposal is to consider the meta-conditions for the scientific proposal. This is an important aspect of scientific reasoning, and it may involve considering the reliability of sources, and the believability, plausibility, or quality of information used. The quality of putative scientific knowledge derived from experiments may be questionable in a number of ways based on the quality of the experimental environment, the quality of the starting materials, the nature of any subjects being studied, and the nature of the scientific methodology. The quality may also be questionable in terms of the interpretation of the scientific results, so that cause-effect relationships are claimed for results that should be interpreted as coincidence. Alternatively, incorrect statistical techniques may have been used. So a scientific proposal can be qualified in a number of ways, and arguments against these qualifications can then be represented.

Definition 9. *Let A_j be an undercut for A_i. A_j is a* **meta-condition violation** *of A_i iff there is a scientific proposal $f_i \in \mathsf{Support}(A_i)$ and there is a ground version γ' of $\mathsf{Metacondition}(f_i)$ such that $\mathsf{Support}(A_i) \setminus \{f_i\} \vdash \gamma'$ and $\mathsf{Support}(A_j) \cup \{\gamma'\}$ is inconsistent.*

So an argument is subject to a meta-condition violation when the support of the argument includes a scientific proposal and there is an undercut that negates the meta-condition of the scientific proposal. An illustration of a meta-condition violation is given in Example 7.

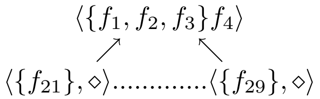
A second way to reflect caution in a scientific proposal is to consider exceptions. As formalized next, an argument is subject to an exception violation when the support of the argument includes a scientific proposal and there is an undercut which has a support that negates the tentative scientific statement for the scientific proposal. As a result, since the consequent is a tentative scientific statement, the ground atoms satisfy the antecedent but negate the consequent of the tentative scientific statement as illustrated in Example 8.

Definition 10. *Let A_j be an undercut for A_i. A_j is an* **exception violation** *of A_i iff $\mathsf{Support}(A_j)$ contains only ground formulae and there is a scientific proposal $f_i \in \mathsf{Support}(A_i)$ such that $\mathsf{Support}(A_j) \cup \{\mathsf{Proposal}(f_i)\}$ is inconsistent.*

Example 8. Consider f_4 given in Example 1. Suppose we have nine exceptions f_{21},..,f_{29} as follows.

$$\begin{array}{ll}
f_{21} & (\texttt{healthy}(\texttt{patient33}) \land \texttt{under75}(\texttt{patient33}) \\
 & \quad \land \texttt{treatment}(\texttt{patient33}, \texttt{p237}, \texttt{50mg}, \texttt{daily}) \\
 & \quad \land \neg \texttt{decreaseBloodCholesterol}(\texttt{patient33}))) \\
\vdots & \\
f_{29} & (\texttt{healthy}(\texttt{patient89}) \land \texttt{under75}(\texttt{patient89}) \\
 & \quad \land \texttt{treatment}(\texttt{patient89}, \texttt{p237}, \texttt{50mg}, \texttt{daily}) \\
 & \quad \land \neg \texttt{decreaseBloodCholesterol}(\texttt{patient89})))
\end{array}$$

Then we have the argument tree


$\langle\{f_1, f_2, f_3\}f_4\rangle$
$\langle\{f_{21}\}, \diamond\rangle$............$\langle\{f_{29}\}, \diamond\rangle$

A problem with this type of violation is that there may be a significant number of exceptions of the same form, and so we may wish to abbreviate the information we have about these exceptions. To support this, a useful conservative extension of the first-order language is qualified statements. These allow us to represent a specific set of examples for which a general statement holds.

Definition 11. *A* **qualified statement** *is a formula of the following form* $\forall x \in \{t_1, .., t_n\}.\alpha$*, where* $\{t_1, .., t_n\}$ *is a set of ground terms, and* α *is a formula.*

Definition 12. *We extend the* $\vdash$ *consequence relation with the following holding for all formulae* α *where* $\alpha[x/t_i]$ *denotes the grounding of all free occurrences of the* x *variable by the ground term* t_i*.*

$$\vdash (\forall x \in \{t_1, .., t_n\}.\alpha) \leftrightarrow (\alpha[x/t_1] \wedge .. \wedge \alpha[x/t_n])$$

Example 9. Let $\Delta = \{\forall x \in \{a, b, c\}.(\forall y \in \{c, d\}.(p(x, y) \rightarrow q(x, a))\}$. Hence, $\Delta \vdash p(b, c) \rightarrow q(b, a)$. If Δ' comprises the formulae below, then $\mathsf{Cn}(\Delta) = \mathsf{Cn}(\Delta')$.

$$\begin{array}{cc}
p(a, c) \rightarrow q(a, a) & p(b, c) \rightarrow q(b, a) \\
p(c, c) \rightarrow q(c, a) & p(a, d) \rightarrow q(a, a) \\
p(b, d) \rightarrow q(b, a) & p(c, d) \rightarrow q(c, a)
\end{array}$$

Qualified statements are useful shorthand for a set of statements. Also if $\Delta \vdash \forall x \in X.\alpha$ and $X' \subseteq X$, then $\Delta \vdash \forall x \in X'.\alpha$.

Example 10. Let us denote f_{30} by the following formula.

$$\begin{array}{l}
\forall \mathtt{x} \in \{\mathtt{patient33}, .., \mathtt{patient89}\}. \\
\qquad\qquad (\mathtt{healthy(x)} \wedge \mathtt{under75(x)} \\
\qquad\qquad\qquad \wedge \mathtt{treatment(x, p237, 50mg, daily)} \\
\qquad\qquad\qquad \wedge \neg \mathtt{decreaseBloodCholesterol(x)}))
\end{array}$$

Using f_1, f_2, and f_3, from Example 1, with f_{30}, the following is an argument tree for f_4.

$$\begin{array}{c}
\langle \{f_1, f_2, f_3\}, f_4 \rangle \\
\uparrow \\
\langle \{f_{30}\}, \Diamond \rangle
\end{array}$$

A third way of expressing caution in a scientific proposal is to identify conflicts with the established scientific knowledge used. As discussed earlier, there are numerous inconsistencies of various kinds in the established literature, and so even though a scientific statement may be assumed to be part of the established knowledge, it does not necessarily mean it is absolutely correct, and such conflicts may need to be highlighted when they are relevant to a scientific proposal under consideration. As formalized next, an argument is subject to presupposition violation when there is a counterargument that negates a scientific statement used in the support of the argument.

Definition 13. *Let* A_j *be an undercut for* A_i*.* A_j *is a* **presupposition violation** *of* A_i *iff there is a scientific statement* $f_i \in \mathsf{Support}(A_i)$ *such that* $\mathsf{Support}(A_j) \cup \{f_i\}$ *is inconsistent.*

Example 11. For the formula below, h_1 is a scientific proposal, h_2 is an established piece of scientific knowledge, and h_3 and h_4 are subsidiary formulae.

$$\begin{aligned} h_1\ &\forall\texttt{x}.(\texttt{validDrugTrial}(\texttt{trial990}) \rightarrow \\ &\qquad(\texttt{decreaseChronicAnxiety}(\texttt{x}) \rightarrow \texttt{increasedLifeExpectancy}(\texttt{x})))\end{aligned}$$

$$h_2\ \forall\texttt{x}.(\texttt{treatment}(\texttt{x}, \texttt{daloxopin}, \texttt{4mg}, \texttt{daily}) \rightarrow \texttt{decreaseChronicAnxiety}(\texttt{x}))$$

$$h_3\ \texttt{validDrugTrial}(\texttt{trial990}))$$

$$h_4\ \forall\texttt{x} \in \{\texttt{patient1}, .., \texttt{patient241}\}.(\texttt{treatment}(\texttt{x}, \texttt{daloxopin}, \texttt{4mg}, \texttt{daily})$$

Assuming $\{h_1, h_h, h_3, h_4\}$ we obtain h_5 by implication.

$$h_5\ \forall\texttt{x}.(\texttt{decreaseChronicAnxiety}(\texttt{x}) \rightarrow \texttt{increasedLifeExpectancy}(\texttt{x}))$$

In addition, assume we have the formula h_6 that says that it is not the case, for any patient, any dose, or any frequency of treatment, that daloxopin decreases chronic anxiety. This formula is therefore negating some established scientific knowledge used above.

$$h_6\ \forall\texttt{x}, \texttt{y}, \texttt{z}.\neg(\texttt{treatment}(\texttt{x}, \texttt{daloxopin}, \texttt{y}, \texttt{z}) \rightarrow \texttt{decreaseChronicAnxiety}(\texttt{x})$$

From this, we get the following argument tree that reflects the presupposition violation.

$$\begin{tikzcd}
\langle\{h_1, h_2, h_3, h_4\}, h_5\rangle \\
\langle\{h_6\}, \diamond\rangle \arrow[u]
\end{tikzcd}$$

As stated earlier, undercuts can also be undercut by recursion. These undercuts may also include circumstantial undercuts which are undercuts based on special circumstances arising when undertaking the trial or experiment or when drawing up the scientific proposal. For example, an exception may be undercut because it arises from a possibly faulty observation or an incorrect experimental set-up.

5 Degree of Undercut

An argument conflicts with each of its undercuts, by the very definition of an undercut. Now, some may conflict more than others, and some may conflict a little while others conflict a lot. To illustrate, consider the following trees.

$$\begin{tikzcd}
T_1 & T_2 & T_3 \\
\langle\{P(a)\}, P(a)\rangle & \langle\{\forall x.P(x)\}, P(a)\rangle & \langle\{\forall x.P(x)\}, P(a)\rangle \\
\langle\{\neg P(a)\}, \diamond\rangle \arrow[u] & \langle\{\neg P(a)\}, \diamond\rangle \arrow[u] & \langle\{\neg P(b)\}, \diamond\rangle \arrow[u]
\end{tikzcd}$$

All of $T_1, .., T_3$ have $P(a)$ as the conclusion. In T_1, the support for root is $\{P(a)\}$ and the support for the undercut is $\{\neg P(a)\}$. This can be described as a propositional conflict where $P(a)$ is against $\neg P(a)$. In T_2, the support for root is $\{\forall x.P(x)\}$ and the support for the undercut is $\{\neg P(a)\}$. This can be described as equivalent to T_1 since the conflict is only with respect to one grounding of x, viz. the grounding by a. In T_3, the support for the root is $\{\forall x.P(x)\}$ but the support for the undercut is $\{\neg P(b)\}$. This can also be described as equivalent to T_1 since the conflict is only with respect to one grounding of x, viz. the grounding by b.

$$\begin{tikzcd}
T_4 & T_5 & T_6 \\
\langle\{\forall x.P(x)\}, \forall x.P(x)\rangle & \langle\{\forall x.P(x)\}, \forall x.P(x)\rangle & \langle\{\forall x.P(x)\}, \forall x.P(x)\rangle \\
\langle\{\neg P(a)\}, \diamond\rangle \arrow[u] & \langle\{\neg\forall x.P(x)\}, \diamond\rangle \arrow[u] & \langle\{\forall x.\neg P(x)\}, \diamond\rangle \arrow[u]
\end{tikzcd}$$

All of $T_4, .., T_6$ have $\forall x.P(x)$ as the conclusion. In T_4, the support for the root is $\{\forall x.P(x)\}$ and the support for the undercut is $\{\neg P(a)\}$. So this can be described as having the same degree of conflict as T_2. In T_5, the support for the root is $\{\forall x.P(x)\}$ and the support for the undercut is $\{\neg\forall x.P(x)\}$. Since $\neg\forall x.P(x)$ is logically equivalent to $\exists x.\neg P(x)$, the conflict only necessarily involves one grounding for x. Hence, this can also be described as as having the same degree of conflict as T_2. In T_6, the support for the root is $\{\forall x.P(x)\}$ and the support for the undercut is $\{\forall x.\neg P(x)\}$. Here, the conflict is much more substantial since it involves all possible groundings for x.

By these simple examples, we see there is an intuitive difference in the degree of conflict between supports, and hence an intuitive starting point for defining the degree of undercut that an argument has against its parent. This degree of undercut depends on the logical nature of the supports involved. Above we have considered this informally for some examples of monadic literals. In the following, we review a formal conceptualization of this for formulae involving n-predicates and involving logical connectives [3], and then consider how it can used for analysing scientific arguments. For this, the conflict of an argument with each of its undercuts is reflected by a position in an ordering (possibly a partial one) but not necessarily a numerical value in some interval (i.e., orders of magnitude are not necessarily needed).

Definition 14. *A* **degree of undercut** *is a mapping* $\mathsf{Degree} : \varOmega \times \varOmega \to O$ *where* $\langle O, \leq\rangle$ *is some poset such that for* $A_i = \langle\varPhi, \alpha\rangle$ *and* $A_j = \langle\varPsi, \beta\rangle$ *in* $\varOmega$*,*

(1) $\mathsf{Degree}(A_j, A) \leq \mathsf{Degree}(A_i, A)$ *for all* $A \in \varOmega$ *if* $\varPhi \vdash \varPsi$

(2) $\mathsf{Degree}(A_i, A_j)$ *is minimal iff* $\varPhi \cup \varPsi \nvdash \bot$

The last clause in Definition 14 means that $\mathsf{Degree}(A, A')$ is minimal when A and A' are two arguments which do not conflict with each other (so, none is an undercut of the other, as Degree is rather a degree of conflict but it is called degree of undercut here because we are only interested in its value when A' is an undercut of A). Definition 14 allows for many possibilities, leaving you to choose a suitable mapping.

We now introduce labelled argument trees. I.e., we label each arc with the degree of undercut. In the rest of the paper, we assume that O is the interval $[0, 1]$.

Definition 15. *A* **labelled argument tree** *is an argument tree such that if A_j is a child of A_i in the argument tree, then the arc from A_j to A_i is labelled with* $\mathsf{Degree}(A_i, A_j)$*.*

Example 12. A labelled argument tree for $\forall x.\alpha[x]$ is:

$$\begin{tikzcd}
& \langle\{\forall x.\alpha[x]\}, \forall x.\alpha[x]\rangle & \\
\langle\{\neg\alpha[a]\}, \Diamond\rangle \arrow[ur, "1/n"] & & \langle\{\neg\alpha[b_1] \wedge \ldots \wedge \neg\alpha[b_m]\}, \Diamond\rangle \arrow[ul, "m/n"']
\end{tikzcd}$$

From now on, n is some reasonable upper bound for the size of the universe of discourse (it is supposed to be finite).

One conceptualization for degree of undercut is based on Herbrand Interpretation. For the rest of the paper, we assume that the non-logical language for Δ is restricted to predicate, variable, and constant symbols, and so function symbols are not used. We also assume that Δ includes at least one constant symbol, and normally, numerous constant symbols. Note, there are other conceptualizations of degree of undercut where we do not restrict ourselves to a finite universe, and can use an unrestricted first-order classical language [4].

Definition 16. *Let Π be the set of ground atoms that can be formed from the predicate symbols and constant symbols used in Δ. Π is the* **base** *for Δ. Each $w \subseteq \Pi$ is an* **interpretation** *s.t. each atom in w is assigned true and each atom in $\Pi \setminus w$ is assigned false. For a set of formulae X, let $M(X, \Pi)$ be the set of* **models** *of X that are in $\wp(\Pi)$. So $M(X, \Pi) = \{w \models \wedge X \mid w \in \wp(\Pi)\}$ where $\models$ is classical satisfaction.*

Example 13. Let $X = \{\ q(b) \wedge q(c),\ \neg r(c),\ \forall x.p(x),\ \exists x.(r(x) \wedge q(x))\ \} \subseteq \Delta$ and so $\Pi = \{p(a), p(b), p(c), q(a), q(b), q(c), r(a), r(b), r(c)\}$. Hence $M(X, \Pi)$ contains exactly the following models.

$$\begin{array}{l}
\{p(a), p(b), p(c), q(a), q(b), q(c), r(a), r(b)\} \\
\{p(a), p(b), p(c), q(b), q(c), r(a), r(b)\} \\
\{p(a), p(b), p(c), q(a), q(b), q(c), r(a)\} \\
\{p(a), p(b), p(c), q(a), q(b), q(c), r(b)\} \\
\{p(a), p(b), p(c), q(b), q(c), r(b)\}
\end{array}$$

We now recall the definition for Dalal distance for comparing a pair of models which is the Hamming distance between the two models [8].

Definition 17. *Let $w_i, w_j \in \wp(\Pi)$. The* **Dalal distance** *between w_i and w_j, denoted* $\mathsf{Dalal}(w_i, w_j)$*, is the difference in the number of atoms assigned true:*

$$\mathsf{Dalal}(w_i, w_j) = |w_i - w_j| + |w_j - w_i|$$

To evaluate the conflict between two theories, we take a pair of models, one for each theory, such that the Dalal distance is minimized. The degree of conflict is this distance divided by the maximum possible Dalal distance between a pair of models (i.e. $\log_2$ of the total number of models in $\wp(\Pi)$ which is $|\Pi|$).

Definition 18. *For $X, Y \subseteq \Delta$ s.t. $X \not\vdash \bot$ and $Y \not\vdash \bot$, let* $\mathsf{Distances}(X, Y, \Pi)$ *be*

$$\{\mathsf{Dalal}(w_x, w_y) \mid w_x \in M(X, \Pi) \text{ and } w_y \in M(Y, \Pi)\}$$

The **degree of conflict***, denoted* $\mathsf{Conflict}(X, Y, \Pi)$*, is:*

$$\mathsf{Conflict}(X, Y, \Pi) = \frac{\mathsf{Min}(\mathsf{Distances}(X, Y, \Pi))}{|\Pi|}$$

Example 14. Let $\Pi = \{p(a), p(b), p(c), q(a), q(b), q(c), r(a), r(b), r(c)\}$.

$$\begin{array}{l} \mathsf{Conflict}(\{\forall x.p(x)\}, \{\exists x.\neg p(x)\}, \Pi) = 1/9 \\ \mathsf{Conflict}(\{\forall x.p(x)\}, \{\neg(p(a) \vee p(b))\}, \Pi) = 2/9 \\ \mathsf{Conflict}(\{\forall x.p(x)\}, \{\forall x.\neg p(x)\}, \Pi) = 3/9 \end{array}$$

For $X, Y \subseteq \Delta$, such that $X \not\vdash \bot$ and $Y \not\vdash \bot$, we can show the following: (1) $0 \leq \mathsf{Conflict}(X, Y, \Pi) \leq 1$; (2) $\mathsf{Conflict}(X, Y, \Pi) = \mathsf{Conflict}(Y, X, \Pi)$; and (3) $\mathsf{Conflict}(X, Y, \Pi) = 0$ iff $X \cup Y \not\vdash \bot$.

Definition 19. *Let $A_i = \langle \Phi, \alpha \rangle$ and let $A_j = \langle \Psi, \beta \rangle$ be arguments. The* **Dalal-Herbrand degree of undercut** *by A_j for A_i, denoted* $\mathsf{Degree}_{dh}(A_i, A_j, \Pi)$*, is* $\mathsf{Conflict}(\Phi, \Psi, \Pi)$.

Clearly, if A_i is an undercut for A_j, then $\mathsf{Degree}_{dh}(A_i, A_j, \Pi) > 0$.

Example 15. Let $A_1 = \langle \{\neg \exists x.p(x)\}, \neg \forall x.p(x) \rangle$, $A_2 = \langle \{\exists x.\neg p(x)\}, \neg \forall x.p(x) \rangle$, $A_3 = \langle \{\neg p(a_1)\}, \neg \forall x.p(x) \rangle$, $A_4 = \langle \{\forall x.p(x)\}, p(a_1) \rangle$, and $\Pi = \{p(a_1), ..., p(a_n)\}$.

$$\begin{array}{l} \mathsf{Degree}_{dh}(A_4, A_1, \Pi) = n/n \\ \mathsf{Degree}_{dh}(A_4, A_2, \Pi) = 1/n \\ \mathsf{Degree}_{dh}(A_4, A_3, \Pi) = 1/n \end{array}$$

A scientist can use the degree of undercut to compare arguments and counterarguments. We can regard each argument in a tree as either an **attacking argument** or a **defending argument**. The root is a defending argument. If an argument A_i is a defending argument, then any child A_j of A_i is an attacking argument. If an argument A_j is an attacking argument, then any child A_k of A_j is a defending argument. For a scientific proposal used in the root, a scientist could publish a scientific proposal in the public domain with more confidence, if the undercuts to defending arguments have a low degree of undercut, and the undercuts to attacking arguments have a high degree of undercut.

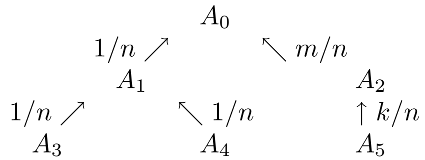
Example 16. Consider the argument tree given in Example 10. Suppose the knowledgebase from which the tree is constructed contains just the formulae f_1, f_2, f_3, f_{30}, together with the following 2479 formulae.

g_1	$(\texttt{healthy(patient1)} \wedge \texttt{under75(patient1)})$
g_2	$(\texttt{healthy(patient2)} \wedge \texttt{under75(patient2)})$
g_3	$(\texttt{healthy(patient3)} \wedge \texttt{under75(patient3)})$
$\vdots$	$\vdots$
g_{2479}	$(\texttt{healthy(patient2479)} \wedge \texttt{under75(patient2479)})$

Hence the Dalal-Herbrand degree of undercut by $\langle\{f_{30}\}, \diamond\rangle$, for $\langle\{f_1, f_2, f_3\}, f_4\rangle$ is 9/2479.

Labelled argument trees provide extra information that leads to a useful abstraction of the original argument tree.

Example 17. Let $A_0, A_1, A_2, ..., A_5$ be arguments, and let $k < n$ and $m < n$ hold. For this, the following is a labelled argument tree.



In the above labelled argument tree, if n is significantly greater than 1, then it may be natural to ignore the left subtree rooted at A_1 and to concentrate on the right-most branch of the tree. If m is close to n, then A_2 is an important undercut of A_0, whereas if it is close to 1, then it may be natural to ignore this branch also.

The tension of an argument tree is the cumulative conflict obtained from all the undercuts in the tree. As tension rises, the more the scientist has to be careful how a new scientific proposal is presented.

Definition 20. *Let T be an argument tree, and let A_r be the root node. The degree of tension in T, denoted* $\mathsf{Tension}(T)$*, is given by the value of* $\mathsf{Retension}(A_r)$*, where for any node A_i in the tree, if A_i is a leaf, then* $\mathsf{Retension}(A_i) = 0$ *otherwise* $\mathsf{Retension}(A_i)$ *is*

$$\sum_{A_j \text{ s.t. } A_j \text{ undercuts } A_i} \mathsf{Retension}(A_j) + \mathsf{Degree}(A_i, A_j, \Pi)$$

Clearly, $\mathsf{Tension}(T) < |\mathsf{Nodes}(T)|$. Furthermore, $|\mathsf{Nodes}(T)| = 1$ if and only if $\mathsf{Tension}(T) = 0$. Tension is maximized when each formula in Δ has to be inconsistent with every other formula, such as $\{\alpha \wedge \beta, \alpha \wedge \neg\beta, \neg\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\}$, so that every argument is an undercut to every other argument.

We conclude this section by sketching another conceptualization of degree of undercut. Here, we assume $\mathcal{O}$ is $\mathbb{N} \cup \{\infty\} \times \mathbb{N} \cup \{\infty\}$ (and so for this paragraph we suspend our general assumption in this section of $\mathcal{O}$ being $[0, 1]$). Informally, for arguments A_i and A_j, the degree of undercut of A_j for A_i is a pair (n, k) where n is the number of situations where the support of A_j is regarded as holding (and thereby justifying the support), and k is the number of situations where the support of A_i is regarded as holding (and thereby justifying the support). Now, consider an argument tree about clinical drug trials, the number of situations where a support holds can be defined in terms of the number of patients involved in the trial. If we consider Example 1, for the argument $\langle\{f_1, f_2, f_3\}, f_4\rangle$, the support is justified by 2479 patients, and if we consider Example 10, for the argument $\langle\{f_{30}\}, \diamond\rangle$ the support is justified by 9 patients. So the degree of undercut is $(9, 2479)$. For supports that use only established scientific knowledge, we use the value ∞ to denote the understanding that the support uses only established scientific knowledge. So an argument with support containing knowledge from a trial involving a 1000 patients that undercuts an argument that uses only established scientific knowledge, the degree of undercut is $(1000, \infty)$. Similarly, for an argument that uses only established scientific knowledge undercutting an argument with support containing knowledge from a trial involving a 1000 patients, the degree of undercut is $(\infty, 1000)$. Finally, for an argument that uses only established scientific knowledge undercutting an argument that uses only established scientific knowledge, the degree of undercut is (∞, ∞).

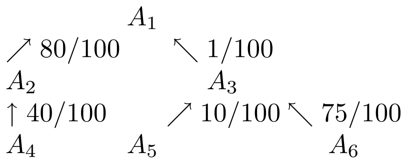
6 Editing Argument Trees

Even for small first-order knowledgebases, the number of arguments generated may be overwhelming for a scientist to be able to study at any one time. To address this problem, we review some proposals for rationalization of argument trees [3, 4] including (1) Pruning arguments that have a degree of undercut that is below a certain threshold; and (2) Merging arguments to create fewer undercuts but without losing vital information. Rationalization is part of a process of editing a set of arguments and counterarguments to allow a scientist to focus on key issues.

For pruning, we choose a threshold for a minimum degree of undercut. If an undercut has a degree of undercut below the threshold, then the undercut is dropped, together with any offspring of that undercut.

Definition 21. *A threshold, denoted τ, is a value in $[0, 1]$ such that if T is an argument tree,* $\mathsf{Prune}(T, \tau)$ *is the* **pruned argument tree** *obtained from T by removing every undercut A_j for an argument A_i if* $\mathsf{Degree}(A_i, A_j, \Pi) \leq \tau$ *and for any undercut removed, all the offspring of that undercut are also removed.*

Example 18. Let T be the following labelled argument tree.



Below, the left argument tree is $\mathsf{Prune}(T, 0.3)$ and the right one is $\mathsf{Prune}(T, 0.5)$.

$$\begin{array}{ll} A_1 & A_1 \\ \uparrow 80/100 & \uparrow 80/100 \\ A_2 & A_2 \\ \uparrow 40/100 & \\ A_4 & \end{array}$$

So pruning of argument trees allows us to focus our attention on the most conflicting undercuts.

Proposition 4. *For $i \in [0,1]$, if $T' = \mathsf{Prune}(T, i)$ then $\mathsf{Tension}(T') \leq \mathsf{Tension}(T)$ and $|\mathsf{Nodes}(T')| \leq |\mathsf{Nodes}(T)|$ and $\mathsf{Depth}(T') \leq \mathsf{Depth}(T)$.*

Also, $\mathsf{Prune}(T, 0) = T$ and $\mathsf{Prune}(T, 1)$ returns a tree containing just the root of T. For all $i \in [0,1]$, if T is a canonical argument tree, then $\mathsf{Prune}(T, i)$ is a canonical argument tree. However, if T is a complete argument tree, then $\mathsf{Prune}(T, i)$ is not necessarily a complete argument tree.

For merging, we use the following notion of compression which combines arguments without loss of essential information. Compression merges siblings in order to reduce the number of arguments and to reduce the "redundancy" arising by having numerous similar arguments or logically equivalent arguments, and to make appropriate "simplifications" of the syntax of some arguments.

Definition 22. *Let T_1 and T_2 be argument trees. T_2 is a* **compression** *of T_1 iff there is a surjection $G : \mathsf{Nodes}(T_1) \to \mathsf{Nodes}(T_2)$ such that for all $B \in \mathsf{Nodes}(T_2)$,*

$$\mathsf{Cn}(\mathsf{Support}(B)) = \mathsf{Cn}\Big(\bigcup_{A \in G^{-1}(B)} \mathsf{Support}(A)\Big)$$

We call G the compression function.

The argument tree in Example 10 is a compression of the argument tree in Example 8. Logical simplification of supports of arguments, as illustrated in the example below, may also be useful in some circumstances. Such simplifications may be important in focussing on the main issues, and removal of less relevant concepts.

Example 19. T_3 is a compression of T_2:

$$\begin{tikzcd}[column sep=small]
& T_2 & & T_3 \\
& \langle\{\forall x.Px\}, \forall x.Px\rangle & & \langle\{\forall x.Px\}, \forall x.Px\rangle \\
\langle \neg Pa \vee \neg Pb, \diamond\rangle \arrow[ur] & & \langle \neg Pa \wedge \neg Pb, \diamond\rangle \arrow[ul] & \langle \neg Pa \wedge \neg Pb, \diamond\rangle \arrow[u]
\end{tikzcd}$$

while each of T_2 and T_3 is a compression of T_1:

$$\begin{tikzcd}[column sep=tiny]
& & & T_1 & & & \\
& & & \langle\{\forall x.Px\}, \forall x.Px\rangle & & & \\
\langle \neg Pa \vee \neg Pb, \diamond\rangle \arrow[urrr] & & \langle \neg Pa, \diamond\rangle \arrow[ur] & & \langle \neg Pb, \diamond\rangle \arrow[ul] & & \langle \neg Pa \wedge \neg Pb, \diamond\rangle \arrow[ulll]
\end{tikzcd}$$

Proposition 5. *If* T' *is a compression of* T*, then* $\mathsf{Tension}(T') \leq \mathsf{Tension}(T)$ *and* $|\mathsf{Nodes}(T')| \leq |\mathsf{Nodes}(T)|$ *and* $\mathsf{Depth}(T') = \mathsf{Depth}(T)$*.*

Compression is not necessarily unique, and there are limits to compression, for example when an argument tree is a chain, and when all pairs of siblings have supports that are mutually contradictory. If compression is restricted to just replacing ground formulae with qualified formulae, then the tension is constant. Alternatively, we may choose to just use compressions that do not change the tension. For more details on compression, and for alternatives, see [3, 4].

A presentation of arguments and counterarguments can also be edited in order to improve the impact of the argumentation [12], and/or to increase the believability of the argumentation [11], from the perspective of the intended audience of argumentation.

For increasing the impact of argumentation, we have developed an evaluation of arguments in terms of how the arguments resonate with the intended audience of the arguments. For example, if a scientist wants to present results from a research project, the arguments used would depend on what is important to the audience: Arguments based on the potential economic benefits of the research would resonate better with an audience from the business community and from the funding agencies, whereas arguments based on the scientific results would resonate better with an audience of fellow scientists. By analysing the resonance of arguments, we can prune argument trees to raise their impact for an audience.

For increasing the believability of argumentation, we have developed a model-theoretic evaluation of the believability of arguments. This extension assumes that the beliefs of a typical member of the audience for argumentation can be represented by a set of classical formulae (a beliefbase). We compare a beliefbase with each argument to evaluate the empathy (or similarly the antipathy) that an agent has for the argument. On the basis of believability, a scientist may wish to ignore arguments for which the audience has antipathy.

The use of pruning, of rationalization, and of selectivity based on raising impact and optimizing believability, is part of a trend to consider the audience in argumentation, and present constellations of arguments and counterarguments that are appropriate for the audience. For example, formalising persuasion has a role in modelling legal reasoning [2].

7 Discussion

The primary aim of this paper has been to provide a framework for presenting scientific arguments and counterarguments based on first-order predicate logic. We can view the framework in this paper as a specification for a decision-support system for scientists to evaluate new scientific proposals. To use it, a scientist would be responsible for adding the relevant scientific knowledge together with the scientific proposal of interest. The decision-support system would then construct the labelled argument trees. Scientists are a user community who may be highly amenable to learning and using predicate logic to use this system. Alternatively, we may need to look towards developments in natural language processing for translating free text into logical formulae.

One of the key advantages of undertaking meta-analysis of scientific knowledge using logic-based argumentation is that when we do not have access to all the original data, we need to deal with the arguments that can be constructed from the publically available information. Consider for example comparing clinical trials undertaken at different hospitals where it may be difficult to have access to all the primary data and/or there may be heterogeneity arising from differing protocols or differing usages of language.

Another way of looking at this is that often the results of an experiment can be captured by a conditional probability statement $P(\alpha \mid \beta)$. This says that the proportion of examples that meet condition β also meet condition α. So a conditional probability statement also captures the proportion of counterexamples which is given by $P(\neg\alpha \mid \beta)$. However, dealing with conditional probabilities cannot be easily extended to dealing with established scientific knowledge, to dealing with exceptions to exceptions, or to dealing with conflicting information, without recourse to a much more comprehensive knowledge of the total probability distribution. This is often impractical or impossible. Scientists do not normally have access to the full experimental data for established scientific knowledge. They normally only have access to the universal statements as an abstraction. So representing conditional probability statements of the form $P(\alpha \mid \beta)$ by statements of the form $\beta \rightarrow \alpha$ when the probability value is greater than say 0.9, is an efficient format. We can reason with the logical formulae using argumentation and represent exceptions by counterarguments. Moreover, we can directly represent inconsistencies in the established scientific knowledge.

Scientific knowledge can also be compared with the commonly considered usage of a default (or defeasible) knowledge. It is noteworthy that human practical reasoning relies much more on exploiting default information than on a myriad of individual facts. Default knowledge tends to be less than 100% accurate, and so has exceptions [5]. Nevertheless it is intuitive and advantageous to resort to such defaults and therefore allow the inference of useful conclusions, even if it does entail making some mistakes as not all exceptions to these defaults are necessarily known. Furthermore, it is often necessary to use default knowledge when we do not have sufficient information to allow us to specify or use universal laws that are always valid. This paper raise an opportunity to revisit the notion of default knowledge, and consider its relevance to scientific knowledge.

The secondary aim of the paper has been to extend logic-based proposals for argumentation with techniques for first-order argumentation. Degree of undercut, labelled argument trees/graphs, and pruning and compressing arguments, could be adapted for other logic-based proposals such as [10, 1, 9].

References

1. L Amgoud and C Cayrol. A model of reasoning based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–216, 2002.
2. T Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
3. Ph Besnard and A Hunter. Practical first-order argumentation. In *Proc. of the 20th National Conference on Artificial Intelligence (AAAI'2005)*, pages 590–595. MIT Press, 2005.
4. Ph Besnard and A Hunter. *Elements of Argumentation*. MIT Press, 2006. (in preparation).
5. G Brewka, J Dix, and K Konolige. *Non-monotonic Reasoning: An Overview*. CLSI Publications, 1997.
6. S Buckingham Shum. Modelling naturalistic argumentation in research literatures: Representation and interaction design issues. *International Journal Intelligent Systems*, 2005. Special Issue on Computational Modelling of Naturalistic Argumentation.
7. C Chesnevar, A Maguitman, and R Loui. Logical models of argument. *ACM Computing Surveys*, 32:337–383, 2001.
8. M Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'88)*, pages 3–7. MIT Press, 1988.
9. P Dung, R Kowalski, and F Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 2005. (in press).
10. A Garca and G Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4:95–138, 2004.
11. A Hunter. Making argumentation more believable. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'2004)*, pages 269–274. MIT Press, 2004.
12. A Hunter. Towards higher impact argumentation. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'2004)*, pages 275–280. MIT Press, 2004.
13. R Loui. A mathematical comment on the fundamental difference between legal theory formation and scientific theory formation. In *Fourth International Workshop on Computational Models of Scientific Reasoning and Applications*, 2005.
14. H Prakken and G Vreeswijk. Logical systems for defeasible argumentation. In D Gabbay, editor, *Handbook of Philosophical Logic*. Kluwer, 2000.
15. S Teufel and M Moens. Summarizing scientific articles – experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4), 2002.

Towards a Formal Framework for the Search of a Consensus Between Autonomous Agents

Leila Amgoud, Sihem Belabbes, and Henri Prade

IRIT - CNRS, 118 route de Narbonne
31062 Toulouse Cedex 9, France
{amgoud, belabbes, prade}@irit.fr

Abstract. This paper aims at proposing a general formal framework for dialogue between autonomous agents which are looking for a common agreement about a collective choice. The proposed setting has three main components: the agents, their reasoning capabilities, and a protocol. The agents are supposed to maintain beliefs about the environment and the other agents, together with their own goals. The beliefs are more or less certain and the goals may not have equal priority. These agents are supposed to be able to make decisions, to revise their beliefs and to support their points of view by arguments. A general protocol is also proposed. It governs the high-level behaviour of interacting agents. Particularly, it specifies the legal moves in the dialogue. Properties of the framework are studied. This setting is illustrated on an example involving three agents discussing the place and date of their next meeting.

Keywords: Argumentation, Negotiation.

1 Introduction

Roughly speaking, negotiation is a process aiming at finding some compromise or consensus between two or several agents about some matters of collective agreement, such as pricing products, allocating resources, or choosing candidates. Negotiation models have been proposed for the design of systems able to bargain in an optimal way with other agents for, e.g., buying or selling products in e-commerce [6].

Different approaches to automated negotiation have been investigated [11], including *game-theoretic* approaches (which usually assume complete information and unlimited computation capabilities), *heuristic-based* approaches which try to cope with these limitations, and *argumentation-based* approaches [3, 1, 10, 8, 7] which emphasize the importance of exchanging information and explanations between negotiating agents in order to mutually influence their behaviors (e.g. an agent may concede a goal having a small priority). Indeed, the two first types of settings do not allow for the addition of information or for exchanging opinions about offers. Integrating argumentation theory in negotiation provides a good means for supplying additional information and also helps agents to convince each other by adequate arguments during a negotiation dialogue.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 264–278, 2006.

In the present work, we consider agents having knowledge about the environment graded in certainty levels and preferences expressed under the form of more or less important goals. Their reasoning model will be based on an argumentative decision framework, as the one proposed in [5] in order to help agents making decisions about what to say during the dialogue, and to support their behavior by founded reasons, namely "safe arguments". We will focus on negotiation dialogues where autonomous agents try to find a joint compromise about a collective choice that will satisfy at least all their most important goals, according to their most certain pieces of knowledge.

The aim of this paper is to propose a general and formal framework for handling such negotiation dialogues. A protocol specifying rules of interaction between agents is proposed. As the agents negotiate about a set of offers in order to choose the best one from their common point of view, it is assumed that the protocol is run, at most, as many times as there are offers. Indeed, each run of the protocol consists of the discussion of an offer by the agents. If that offer is accepted by all the agents, then the negotiation ends successfully. Otherwise, if at least one agent rejects it strongly and doesn't revise its beliefs in the light of new information, the current offer is (at least temporarily) eliminated and a new one is discussed.

We take an example to illustrate our proposed framework. It consists of three human agents trying to set a date and a place for organizing their next meeting. Thus the offers allow for multiple components (date and place). For simplicity reasons, we consider them as combined offers so that if an agent has a reason to refuse an element of a given offer, it refuses the whole offer. One of the agents starts the dialogue by proposing an offer which can be accepted or rejected. The negotiation goes on until a consensus is found, or stops if it is impossible to satisfy all the most important goals of the agents at the same time.

The remainder of this paper is organized as follows: in section 2 we define the mental states of the agents representing their beliefs and goals. In section 3 we present the argumentative decision framework capturing their reasoning capabilities. Section 4 describes a protocol for multi-agent negotiation dialogues. Section 5 illustrates the argued-decision based approach on an example dealing with the choice of a place and a date to organize a meeting. Section 6 concludes the paper and outlines some possible future work.

2 Mental States and Their Dynamics

As said before, it is supposed that the mental states of each agent are represented by bases modeling beliefs and goals graded in terms of certainty and of importance respectively. Following [4, 12], each agent is equipped with $(2n)$ bases, where n is the number of agents taking part to the negotiation.

Let $\mathcal{L}$ be a propositional language and $Wff(\mathcal{L})$ the set of well-formed formulas built from $\mathcal{L}$. Each agent a_i has the following bases:

$\mathcal{K}_i$ $= \{(k^i_p, \rho^i_p), p = 1, s_k\}$ where $k^i_p \in Wff(\mathcal{L})$, is a knowledge base gathering the information the agent has about the environment. The beliefs can be less or more certain. They are associated with certainty levels ρ^i_p.

$\mathcal{G}_i$ $= \{(g^i_q, \lambda^i_q), q = 1, s_g\}$ where $g^i_q \in Wff(\mathcal{L})$, is a base of goals to pursue. These can have different priority degrees, represented by λ^i_q.

$\mathcal{GO}^i_j$ $= \{(go^i_{r,j}, \gamma^i_{r,j}), r = 1, s_{go}(j)\}$, where $j \neq i$, $go^i_{r,j} \in Wff(\mathcal{L})$, are $(n-1)$ bases containing what the agent a_i believes the goals of the other agents a_j are. Each of these goals is supposed to have a priority level $\gamma^i_{r,j}$.

$\mathcal{KO}^i_j$ $= \{(ko^i_{t,j}, \delta^i_{t,j}), t = 1, s_{ko}(j)\}$ where $j \neq i$, $ko^i_{t,j} \in Wff(\mathcal{L})$, are $(n-1)$ bases containing what the agent a_i believes the knowledge of the other agents a_j are. Each of these beliefs has a certainty level $\delta^i_{t,j}$.

This latter base is useful only if the agents intend to simulate the reasoning of the other agents. In negotiation dialogues where agents are trying to find a common agreement, it is more important for each agent to consider the beliefs that it has on the other agents'goals rather than those on their knowledge. Indeed, a common agreement can be more easily reached if the agents check that their offers may be consistent with what they believe are the goals of the others. So in what follows, we will omit the use of the bases $\mathcal{KO}^i_j$.

The different certainty levels and priority degrees are assumed to belong to a unique linearly ordered scale T with maximal element denoted by 1 (corresponding to total certainty and full priority) and a minimal element denoted by 0 corresponding to the complete absence of certainty or priority. m will denote the order-reversing map of the scale. In particular, $m(0) = 1$ and $m(1) = 0$.

We shall denote by $\mathcal{K}^*$ and $\mathcal{G}^*$ the corresponding sets of classical propositions when weights are ignored.

3 Argued Decisions

Recently, Amgoud and Prade [5] have proposed a formal framework for making decisions under uncertainty on the bases of arguments that can be built in favor or against a possible choice. Such an approach has two obvious merits. First, decisions can be more easily explained. Moreover, argumentation-based decision is maybe closer to the way humans make decisions than approaches requiring explicit utility functions and uncertainty distributions. Decisions for an agent are computed from stratified knowledge and preference bases in the sense of Section 2. This approach distinguishes between a *pessimistic* attitude, which focuses on the existence of strong arguments that support a decision, and an *optimistic* one, which concentrates on the absence of strong arguments against a considered choice. This approach can be related to the estimation of qualitative pessimistic and optimistic expected utility measures. Indeed, such measures can be obtained from a qualitative plausibility distribution and a qualitative preference profile that can be associated with a stratified knowledge base and with a stratified set of goals [5].

In this paper, we only use the syntactic counterpart of these semantical computations in terms of distribution and profile (which has been proved to be equivalent for selecting best decisions), under its argumentative form. This syntactic approach is now recalled and illustrated on an example.

The idea is that a decision is justified and supported if it leads to the satisfaction of at least the most important goals of the agent, taking into account the most certain part of knowledge. Let $\mathcal{D}$ be the set of all possible decisions, where a decision d is a literal.

Definition 1 (Argument PRO). *An* argument in favor of *a decision d is a triple A = $<S, C, d>$ such that:*

- $d \in \mathcal{D}$
- $S \subseteq \mathcal{K}^*$ *and* $C \subseteq \mathcal{G}^*$
- $S \cup \{d\}$ *is consistent*
- $S \cup \{d\} \vdash C$
- *S is minimal and C is maximal (for set inclusion) among the sets satisfying the above conditions.*

S = $Support(A)$ is the support *of the argument, C = Consequences(A) its* consequences *(the goals which are reached by the decision d) and d = $Conclusion(A)$ is the conclusion of the argument. The set $\mathcal{A}_P$ gathers all the arguments which can be constructed from $<\mathcal{K}, \mathcal{G}, \mathcal{D}>$.*

Due to the stratification of the bases $\mathcal{K}_i$ and $\mathcal{G}_i$, arguments in favor of a decision are more or less strong for i.

Definition 2 (Strength of an Argument PRO). *Let A = $< S, C, d >$ be an argument in $\mathcal{A}_P$.*
The strength *of A is a pair $<Level_P(A), Weight_P(A)>$ such that:*

- *The* certainty level *of the argument is $Level_P(A)$ = $min\{\rho_i \mid k_i \in S$ and $(k_i, \rho_i) \in \mathcal{K}\}$. If $S = \emptyset$ then $Level_P(A)$ = 1.*
- *The* degree of satisfaction *of the argument is $Weight_P(A)$ = $m(\beta)$ with β = $max\{\lambda_j \mid (g_j, \lambda_i) \in \mathcal{G}$ and $g_j \notin C\}$. If $\beta = 1$ then $Weight_P(A) = 0$ and if C = $\mathcal{G}^*$ then $Weight_P(A) = 1$.*

Then, strengths of arguments make it possible to compare pairs of arguments as follows:

Definition 3. *Let A and B be two arguments in $\mathcal{A}_P$. A is* preferred *to B, denoted $A \succeq_P B$, iff $min(Level_P(A), Weight_P(A)) \geq min(Level_P(B), Weight_P(B))$.*

Thus arguments are constructed in favor of decisions and those arguments can be compared. Then decisions can also be compared on the basis of the relevant arguments.

Definition 4. *Let $d, d' \in \mathcal{D}$. d is* preferred *to d', denoted $d \triangleright_P d'$, iff $\exists$ $A \in \mathcal{A}_P$, $Conclusion(A) = d$ such that $\forall$ $B \in \mathcal{A}_P$, $Conclusion(B) = d'$, then $A \succeq_P B$.*

This decision process is pessimistic in nature since it is based on the idea of making sure that the important goals are reached. An optimistic attitude can be also captured. It focuses on the idea that a decision is all the better as there is no strong argument against it.

Definition 5 (Argument CON). *An* argument against *a decision d is a triple* $A = <S, C, d>$ *such that:*

- $d \in \mathcal{D}$
- $S \subseteq \mathcal{K}^*$ *and* $C \subseteq \mathcal{G}^*$
- $S \cup \{d\}$ *is consistent*
- $\forall\ g_i \in C,\ S \cup \{d\} \vdash \neg g_i$
- *S is minimal and C is maximal (for set inclusion) among the sets satisfying the above conditions.*

S = *Support*(A) *is the* support *of the argument,* C = *Consequences(A) its* consequences *(the goals which are not satisfied by the decision d), and d = Conclusion*(A) *its conclusion. The set* $\mathcal{A}_O$ *gathers all the arguments which can be constructed from* $<\mathcal{K}, \mathcal{G}, \mathcal{D}>$.

Note that the consequences considered here are the negative ones. Again, arguments are more or less strong or weak.

Definition 6 (Weakness of an Argument CON). *Let* $A = < S, C, d >$ *be an argument of* $\mathcal{A}_O$.
The weakness *of A is a pair* $<Level_O(A), Weight_O(A)>$ *such that:*

- *The* level *of the argument is* $Level_O(A) = m(\varphi)$ *such that* $\varphi = min\{\rho_i \mid k_i \in S$ *and* $(k_i, \rho_i) \in \mathcal{K}\}$. *If* $S = \emptyset$ *then* $Level_O(A) = 0$.
- *The* degree *of the argument is* $Weight_O(A) = m(\beta)$ *such that* $\beta = max\{\lambda_j$ *such that* $g_j \in C$ *and* $(g_j, \lambda_i) \in \mathcal{G}\}$.

Once we have defined the arguments and their weaknesses, pairs of arguments can be compared. Clearly, decisions for which all the arguments against it are weak will be preferred, i.e. we are interested in the least weak arguments against a considered decision. This leads to the two following definitions:

Definition 7. *Let A and B be two arguments in* $\mathcal{A}_O$. *A is* preferred *to B, denoted* $A \succeq_O B$, *iff* $max(Level_O(A), Weight_O(A)) \geq max(Level_O(B), Weight_O(B))$.

As in the pessimistic case, decisions are compared on the basis of the relevant arguments.

Definition 8. *Let* $d, d' \in \mathcal{D}$. *d is* preferred *to* d', *denoted* $d \rhd_O d'$, *iff* $\exists\ A \in \mathcal{A}_O$ *with Conclusion*$(A) = d$ *such that* $\forall\ B \in \mathcal{A}_O$ *with Conclusion*$(B) = d'$, *then A is preferred to B.*

Let us illustrate this approach using the two points of view (pessimistic and optimistic) on an example about deciding or not to argue in a multiple agent dialogue for an agent which is not satisfied with the current offer.

Example 1. *The knowledge base is* $\mathcal{K} = \{(a \rightarrow suu, 1), (\neg a \rightarrow \neg\ suu, 1), (a \rightarrow \neg aco, 1), (fco \wedge \neg a \rightarrow aco, 1), (sb, 1), (\neg fco \rightarrow \neg aco, 1), (sb \rightarrow fco, \lambda)\}$ $(0 < \lambda < 1)$ *with the intended meaning:*
suu: saying something unpleasant,
fco: other agents in favor of current offer,
aco: obliged to accept the current offer,
a: argue,
sb: current offer seems beneficial for the other agents.
The base of goals is $\mathcal{G} = \{(\neg aco, 1), (\neg suu, \sigma)\}$ *with* $(0 < \sigma < 1)$.
The agent does not like to say something unpleasant, but it is more important not to be obliged to accept the current offer.
The set of decisions is $\mathcal{D} = \{a, \neg a\}$, *i.e., arguing or not.*

There is one argument in favor of the decision 'a': $< \{a \rightarrow \neg aco\}, \{\neg aco\}, a>$. *There is also a unique argument in favor of the decision* '$\neg a'$: $<\{\neg a \rightarrow \neg suu\}, \{\neg suu\}, \neg a>$.

The level of the argument $<\{a \rightarrow \neg aco\}, \{\neg aco\}, a>$ *is 1 whereas its weight is* $m(\sigma)$. *Concerning the argument* $<\{\neg a \rightarrow \neg suu\}, \{\neg suu\}, \neg a>$, *its level is 1 and its weight is m(1) = 0.*

The argument $<\{a \rightarrow \neg aco\}, \{\neg aco\}, a>$ *is preferred to the argument* $<\{\neg a \rightarrow \neg suu\}, \{\neg suu\}, \neg a>$.

¿From a pessimistic point of view, decision a is preferred to the decision $\neg a$ *since* $<\{a \rightarrow \neg aco\}, \{\neg aco\}, a>$ *is preferred to* $<\{\neg a \rightarrow \neg suu\}, \{\neg suu\}, \neg a>$.

Let us examine the optimistic point of view. There is one argument against the decision 'a': $<\{a \rightarrow suu\}, \{\neg suu\}, a>$. *There is also a unique argument against the decision* $\neg a$: $<\{sb, sb \rightarrow fco, fco \wedge \neg a \rightarrow aco\}, \{\neg aco\}, \neg a>$.

The level of the argument $<\{a \rightarrow suu\}, \{\neg suu\}, a>$ *is 0 whereas its degree is* $m(\sigma)$. *Concerning the argument* $<\{sb, sb \rightarrow fco, fco \wedge \neg a \rightarrow aco\}, \{\neg aco\}, \neg a>$, *its level is* $m(\lambda)$, *and its degree is 0.*

Then the comparison of the two arguments amounts to compare $m(\sigma)$ *with* $m(\lambda)$.

The final recommended decision with the optimistic approach depends on this comparison.

This argumentation system will be used to take decisions about the offers to propose in a negotiation dialogue. The following definition is the same as Definition 1 where the decision d is about offers.

Definition 9 (Argument for an offer). *An* argument *in favor of an offer x is a triple* $A = <S, C, x>$ *such that:*

- $x \in X$
- $S \subseteq \mathcal{K}^*$ *and* $C \subseteq \mathcal{G}^*$
- $S(x)$ *is consistent*

- *$S(x) \vdash C(x)$*
- *S is minimal and C is maximal (for set inclusion) among the sets satisfying the above conditions.*

X is the set of offers, $S = Support(A)$, $C = Consequences(A)$ (the goals which are satisfied by the offer x) and $x = Conclusion(A)$. $S(x)$ (resp. $C(x)$) denotes the belief state (resp. the preference state) when an offer x takes place.

Example 2. *The example is about an agent wanting to propose an offer corresponding to its desired place for holidays.*
The set of available offers is $X = \{Tunisia, Italy\}$.
Its knowledge base is:

$\mathcal{K} = \{(Sunny(Tunisia),\ 1),\ (\neg Cheap(Italy),\ \beta), (Sunny(x) \rightarrow Cheap(x),\ 1)\}$.
Its preferences base is: $\mathcal{G} = \{(Cheap(x), 1)\}$.

The decision to take by the agent is whether to offer Tunisia or Italy. Following the last definition, it has an argument in favor of Tunisia:

$A =< \{Sunny(Tunisia), Sunny(x) \rightarrow cheap(x)\},\ cheap(Tunisia), tunisia >$.
It has no argument in favor of Italy (it violates its goal which is very important). So this agent will offer Tunisia.

4 The Negotiation Protocol

4.1 Formal Setting

In this section, we propose a formal protocol handling negotiation dialogues between many agents ($n \geq 2$). Agents having to discuss several offers, the protocol is supposed to be run as many times as there are non-discussed offers, and such that a common agreement is still not found. The agents take turns to start new runs of the protocol and only one offer is discussed at each run.

A negotiation interaction protocol is a tuple ⟨ Objective, Agents, Object, Acts, Replies, Wff-Moves, Dialogue, Result⟩ such that:

Objective is the aim of the dialogue which is to find an acceptable offer.
Agents is the set of agents taking part to the dialogue, $Ag = \{a_0, \ldots, a_{n-1}\}$.
Object is the subject of the dialogue. It is a multi-issue one, denoted by the tuple $\langle O_1, \ldots, O_m \rangle$, $m \geq 1$. Each O_i is a variable taking its values in a set T_i.
Let X be the set of all possible offers, its elements are $x = \langle x_1, \ldots, x_m \rangle$ with $x_i \in T_i$.
Acts is the set of possible negotiation speech acts: *Acts* = {Offer, Challenge, Argue, Accept, Refuse, Withdraw, Say nothing}.
Replies: Acts $\longrightarrow$ Power(Acts), is a mapping that associates to each speech act its possible replies.

- $Replies(Offer)$ = {Accept, Refuse, Challenge}
- $Replies(Challenge)$ = {Argue}
- $Replies(Argue)$ = {Accept, Challenge, Argue}
- $Replies(Accept)$ = {Accept, Challenge, Argue, Withdraw}
- $Replies(Refuse)$ = {Accept, Challenge, Argue, Withdraw}
- $Replies(Withdraw) = \emptyset$

Well-founded moves $=\{M_0,\ldots,M_p\}$ is a set of tuples $M_k = \langle S_k, H_k, Move_k\rangle$, such that:
- $S_k \in Agents$, the agent which plays the move is given by the function $Speaker(M_k) = S_k$.
- $H_k \subseteq Agents \setminus \{S_k\}$, the set of agents to which the move is addressed is given by the function $Hearer(M_k) = H_k$.
- $Move_k = Act_k(c_k)$ is the uttered move where Act_k is a speech act applied to a content c_k.

Dialogue is a finite non-empty sequence of well-founded moves $\mathcal{D} = \{M_0, \ldots, M_p\}$ such that:
- $M_0 = \langle S_0, H_0, offer(x)\rangle$: each dialogue starts with an offer $x \in X$
- $Move_k \neq offer(x), \forall k \neq 0$ and $\forall x \in X$: only one offer is proposed during the dialogue at the first move
- $Speaker(M_k) = a_{k\ modulo\ n}$: the agents take turns during the dialogue.
- $Speaker(M_k) \notin Hearer(M_k)$. This condition forbids an agent to address a move to itself.
- $Hearer(M_0) = a_j, \forall j \neq i$: the agent a_i which utters the first move addresses it to all the agents.
- For each pair of tuples M_k, M_h, $k \neq h$, if $S_k = S_h$ then $Move_k \neq Move_h$. This condition forbids an agent to repeat a move that it has already played.

These conditions guarantee that the dialogue $\mathcal{D}$ is ***non circular***.

Result: $\mathcal{D} \longrightarrow \{success, failure\}$, is a mapping which returns the result of the dialogue.
- $Result(\mathcal{D}) = success$ if the preferences of the agents are satisfied by the current offer.
- $Result(\mathcal{D}) = failure$ if the most important preferences of at least one agent are violated by the current offer.

This protocol is based on dialogue games. Each agent is equipped with a *commitment store* (CS) [9] containing the set of facts it is committed to during the dialogue.

Using the idea introduced in [2] of decomposing the agents' commitments store (CS) into many components, we suppose that each agent's CS has the structure

$$CS = \langle \mathcal{S}, \mathcal{A}, \mathcal{C} \rangle$$

with:

$CS.\mathcal{S}$ contains the offers proposed by the agent and those it has accepted ($CS.\mathcal{S} \subseteq X$),
$CS.\mathcal{A}$ is the set of arguments presented by the agent ($CS.\mathcal{A} \subseteq Arg(\mathcal{L})$), where $Arg(\mathcal{L})$) is the set of all arguments we can construct from $\mathcal{L}$,
$CS.\mathcal{C}$ is the set of challenges made by the agent.

At the first run of the protocol, all the CS are empty. This is not the case when the protocol is run again. Indeed, agents must keep their previous commitments to avoid to repeat what they have already uttered during previous runs of the protocol.

4.2 Conditions on the Negotiation Acts

In what follows, we specify for each act its pre-conditions and post-conditions (effects). For the agents' commitments (CS), we only specify the changes to effect. We suppose that agent a_i addresses a move to the $(n-1)$ other agents.

Offer(x) where $x \in X$. It's the basic move in negotiation. The idea is that an agent chooses an offer x for which there are the strongest supporting arguments (w.r.t. $\mathcal{G}_i$). Since the agent is *cooperative* (it tries to satisfy its own goals taking into account the goals of the other agents), this offer x is the also the one for which there exists no strong argument against it (using $\mathcal{GO}^i_j$ instead of $\mathcal{G}_i$).

Pre-conditions: Among the elements of X, choose x which is preferred to any $x' \in X$ such that $x' \neq x$, in the sense of definition 4, provided that there is no strong argument against the offer x (i.e. with a weakness degree equal to 0) where $\mathcal{G}_i$ is changed into $\mathcal{GO}^i_j$, $\forall j \neq i$ in definition 8.
Post-conditions: $CS.\mathcal{S}_t(a_i) = CS.\mathcal{S}_{t-1}(a_i) \cup \{x\}$.

Challenge(x) where $x \in X$. This move incites the agent which receives it to give an argument in favor of the offer x. An agent asks for an argument when this offer is not acceptable for it and it knows that there are still non-rejected offers.

Pre-conditions: $\exists x' \in X$ such that x' is preferred to x w.r.t. definition 4.
Post-conditions: $CS.\mathcal{C}_t(a_i) = CS.\mathcal{C}_{t-1}(a_i) \cup \{x\}$: the agent a_i which played the move *Challenge*(x) keeps it in its CS.

Challenge(y) where $y \in Wff(\mathcal{L})$. This move incites the agent which receives it to give an argument in favor of the proposition y.

Pre-conditions: There is no condition.
Post-conditions: $CS.\mathcal{C}_t(a_i) = CS.\mathcal{C}_{t-1}(a_i) \cup \{y\}$: the agent a_i which played the move *Challenge*(y) keeps it in its CS.

Argue(S) with $S = \{(k_p, \alpha_p), p = 1, s\} \subseteq \mathcal{K}_i$ is a set of formulas representing the support of an argument given by agent a_i. In [5], it is shown how to compute and evaluate *acceptable* arguments.

Pre-conditions: S is acceptable.
Post-conditions: $CS.\mathcal{A}_t(a_i) = CS.\mathcal{A}_{t-1}(a_i) \cup S$. If S is acceptable (according to the definition given in [5]), the agents a_j revise their base $\mathcal{K}_j$ into a new base $(\mathcal{K}_j)^*(S)$.

Withdraw: An agent can withdraw from the negotiation if it hasn't any acceptable offer to propose.

Pre-conditions: $\forall x \in X$, there is an argument with maximal strength against x, or $(X = \emptyset)$.
Post-conditions: *(Result*$(\mathcal{D})$ *= failure)* and $\forall i, CS_t(a_i) = \emptyset$. As soon as an agent withdraws, the negotiation ends and all the commitment stores are emptied.
We suppose the dialogue ends this way because we aim to find a compromise between the n agents taking part to the negotiation.

Accept(x) where $x \in X$. This move is played when the offer x is acceptable for the agent.

Pre-conditions: The offer x is the most preferred decision in X in the sense of definition 4.
Post-conditions: $CS.\mathcal{S}_t(a_i) = CS.\mathcal{S}_{t-1}(a_i) \cup \{x\}$.
If $x \in CS.\mathcal{S}(a_i)$, $\forall i$, then *Result*$(\mathcal{D})$ = *success*, i.e if all the agents accept the offer x, the negotiation ends with x as compromise.

Accept(S) $S \subset Wff(\mathcal{L})$.

Pre-conditions: S is acceptable for a_i.
Post-conditions: $CS.\mathcal{A}_t(a_i) = CS.\mathcal{A}_{t-1}(a_i) \cup S$.

Refuse(x) where $x \in X$. An agent refuses an offer if it is not acceptable for it.

Pre-conditions: There exists an argument in the sense of definition 5 against x.
Post-conditions: If $\forall a_j, \nexists (S, x)$, i.e. if there not exist any acceptable argument for x then $X = X \setminus \{x\}$. A rejected offer is removed from the set X. *Result*$(\mathcal{D})$ = *failure.*

Say nothing: This move allows an agent to miss its turn if it has already accepted the current offer, or it has no argument to present. This move has no effect on the dialogue.

4.3 Properties of the Negotiation Protocol

Property 1 (Termination). *Any negotiation between n agents managed by our protocol ends, either with* Result$(\mathcal{D})$ = success *or* Result$(\mathcal{D})$ = failure.

Property 2 (Optimal outcome). *If the agents do not misrepresent the preferences of the other agents ($\mathcal{GO}^i_j$), then the compromise found is an offer x which is preferred to any other offer $x' \in X$ in the sense of definition 4, for all the agents.*

5 Example of Deliberative Choice

We illustrate our negotiation protocol through an example of dialogue between three agents: Mary, John and Peter, partners on a common project aiming at setting a town and a date for their next meeting. The negotiation object O is in this case the couple $(Town, Date)$ denoted $\langle t, d\rangle$, where t is for the town and d the date.

Suppose that the set of offers is $X = \{(V, E), (L, S), (V, J)\}$, i.e. the meeting will take part either in Valencia (denoted V), at one of the dates respectively denoted E and J; or in London (denoted L) at the date denoted S.

In what follows, we use the following scale $T = \{a, b, c, d\}$ with the condition $a > b > c > d$. We recall that m is the order reversing map on the scale T such that $m(a) = d$ and $m(b) = c$.

Suppose Mary has the following *beliefs*:

$\mathcal{K}_0$ = {(disposable(V,E), 1), (disposable(t,d) → meet(t,d), 1), (free(V,E),1), (¬ free(L,S) ,1), (disposable(t,J),1)}.

The *goals* of Mary are to meet her partners in any town and at any date, provided that accommodations are free. This can be written: $\mathcal{G}_0$ = {(meet, 1),(free, b)}.

Where "*meet*" is a short for $(meet(V, E) \vee meet(L, S) \vee meet(V, J))$. "*free*" is defined the same way. We use this type of abbreviation in what follows.

Suppose John's beliefs are: $\mathcal{K}_1 = \{(hot(V, d), a)$, $(\neg hot(L, S), 1)$, (disposable $(L, S), 1)$,

$(disposable(t, d) \rightarrow meet(t, d), 1)$, $(meet(V, J) \rightarrow work\_saturday, 1)\}$.

His goals are to meet his partners in any town and at any date, and that this town must be not hot at this date. We write:

$\mathcal{G}_1 = \{(meet, 1), (\neg hot, c)\}$.

Finally we suppose Peter's beliefs are:
$\mathcal{K}_2 = \{(\neg meet(V, E), 1)$, $(\forall d \neq E, meet(V, d), 1)$,
$(disposable(t, d) \rightarrow meet(t, d), 1)$, $(disposable(V, J), b)$,
$(manager, 1)$, $(manager \rightarrow work\_saturday, 1)\}$.

His goals are to meet his partners and to don't work on Saturday. We write: $\mathcal{G}_2 = \{(meet, 1), (\neg work\_saturday, d)\}$.

For simplicity, we suppose that Mary, John and Peter ignore the preferences of each other. This means that $\mathcal{GO}^i_j = \emptyset$, $\forall i, j$.

In what follows, we illustrate the dialogue between the agents and give the moves played by each agent.

First run of the protocol

Mary starts the dialogue by proposing an offer.

Mary: The next meeting should be in Valencia during the conference ECAI. $Offer(V, E)$.
Pre-condition: (V,E) is the most preferred decision for Mary.
Post-condition: $CS.\mathcal{S}(Mary) = \{(V, E)\}$.

John: Why? $Challenge(V, E)$.
Pre-condition: For John, there exists another decision which is preferred to (V,E).
Post-condition: $CS.\mathcal{C}(John) = \{(V, E)\}$.

Peter: What are the advantages? $Challenge(V, E)$.
Pre-condition: For Peter, this decision violates his most important goal.
Post-condition: $CS.\mathcal{C}(Peter) = \{(V, E)\}$.

Mary: I think we can meet as soon as it will be during ECAI. $Argue(meet(V, E))$.
Pre-condition: The argument is acceptable.
Post-condition: $CS.\mathcal{A}(Mary) = \{disposable(V, E), disposable(V, E) \rightarrow meet(V, E)\}$.

John: I refuse Valencia because it is hot. $Argue(hot(V, d))$.
Pre-condition: $\{hot(V, d)\}$ is an acceptable argument.
Post-condition: $CS.\mathcal{A}(John) = \{hot(V, d)\}$.

Peter: For my part, I will not be able to meet you. $Argue(\neg meet(V, E))$.
Pre-condition: $\{\neg meet(V, E)\}$ is an acceptable argument.
Post-condition: $CS.\mathcal{A}(Peter) = \{\neg meet(V, E)\}$.

Mary: Nevertheless the accommodation will be free. $Argue(free(V, E))$.
Pre-condition: $\{free(V, E)\}$ is an acceptable argument.
Post-condition: $CS.\mathcal{A}(Mary) = CS.\mathcal{A}(Mary) \cup \{free(V, E)\}$.

John: It still doesn't fit me. $Refuse(V, E)$.
Pre-condition: the offer violates one of his goals.

Peter: Neither do I. $Refuse(V, E)$.
Pre-condition: the offer violates his most important goal.
Post-condition: *Result($\mathcal{D}$) = failure.*
$X = X \setminus \{(V, E)\}$ and all the CS are emptied except the components of the arguments.

Second run of the protocol: It is started by John.

John: What about London in September ? $Offer(L, S)$.
Pre-condition: (L,S) is the most preferred decision for John.
Post-condition: $CS.\mathcal{S}(John) = \{(L, S)\}$.

Peter: I refuse. $Refuse(L, S)$.
Pre-condition: this offer violates his most important goal.

Mary: John, what are your arguments in favor of your offer ? $Challenge(L, S)$.
Pre-condition: (L,S) is not the preferred decision for Mary.
Post-condition: $CS.\mathcal{C}(Mary) = \{(L, S)\}$.

John: London is not hot and I will be able to meet you.
$Argue(\neg hot(L,S), meet(L,S))$.
Pre-condition: The argument is acceptable.
Post-condition: $CS.\mathcal{A}(John) = CS.\mathcal{A}(John) \cup$
$\{\neg hot(L,S), disposable(L,S), disposable(L,S) \rightarrow meet(L,S)\}$.

Peter: But we have already said that the meeting should be in Valencia ! $Argue(\forall d \neq E, meet(V,d))$.
Pre-condition: $\{\forall d \neq E, meet(V,d)\}$ is an acceptable argument.
Post-condition: $CS.\mathcal{A}(Peter) = CS.\mathcal{A}(Peter) \cup \{\forall d \neq E, meet(V,d)\}$.

Mary: I can't pay the accommodation. $Argue(\neg free(L,S))$.
Pre-condition: $\{\neg free(L,S)\}$ is acceptable.
Post-condition: $CS.\mathcal{A}(Mary) = CS.\mathcal{A}(Mary)$
$\cup \{\neg free(L,S)\}$.

John: I forgot that we have chosen Valencia to organize the meeting. $Accept(\forall d \neq E, meet(V,d))$.
Pre-condition: $\{\forall d \neq E, meet(V,d)\}$ is an acceptable argument.
Post-condition: $CS.\mathcal{A}(John) = CS.\mathcal{A}(John) \cup \{\forall d \neq E, meet(V,d)\}$.
John revises its belief base which becomes: $\mathcal{K}_1$ = {(hot(V,d), 0.9), (¬ hot(L,S), 1), (disposable(L,S), 1), (disposable(t,d) → meet(t,d), 1), (meet(V,J) → work_saturday, 1), (∀ d ≠ E, meet(V,d), 1)}.

Peter: I have nothing more to say. *Say_nothing*.

Mary: I also forgot this. $Accept(\forall d \neq E, meet(V,d))$.
Pre-condition: $\{\forall d \neq E, meet(V,d)\}$ is acceptable.
Post-condition: $CS.\mathcal{A}(Mary) = CS.\mathcal{A}(Mary) \cup \{\forall d \neq E, meet(V,d)\}$.
Mary revises its belief base which becomes:

$\mathcal{K}_0$ = {(disposable(V,E), 1), (disposable(t,d) → meet(t,d), 1), (free(V,E),1), (¬ free(L,S) ,1), (disposable(t,J),1), (∀ d ≠ E, meet(V,d), 1)}.

Result($\mathcal{D}$) = failure, $X = X \setminus \{(L,S)\} = \{(V,J)\}$ and all the CS are emptied except the components of the arguments.

***Third run of the protocol*:** It is Peter's turn to propose an offer.

Peter: What about reorganizing the sessions to satisfy every body? Offer(V,J).
Pre-condition: the decision (V,J) is the most preferred one.
Post-condition: $CS.\mathcal{S}(Peter) = \{(V,J)\}$.

Mary: It fits me. $Accept(V,J)$.
Pre-condition: (V,J) is the most preferred decision for her.
Post-condition: $CS.\mathcal{S}(Mary) = \{(V,J)\}$.

John: Not me ! $Refuse(V,J)$.
Pre-condition: the decision (V,J) violates one of his goals.

Peter: John, what doesn't fit you ?
$Challenge(Refuse(V,J))$.
Pre-condition: There aren't.
Post-condition: $CS.\mathcal{C}(Peter) = \{(V,J)\}$,

Mary: I have nothing to say. *Say_nothing*.

John: If we organize the sessions this way, the managers would have to work on Saturday. $Argue(work\_saturday)$.
Pre-condition: the argument is acceptable.
Post-condition: $CS.\mathcal{A}$(John) = CS.$\mathcal{A}$(John) ∪ {meet(V,J), meet(V,J) → work_saturday}.

Peter: The managers can make the effort of working on Saturday. $Argue(manager, manager \rightarrow work\_saturday)$.
Pre-condition: Peter has an acceptable argument to convince John: $\{manager, manager \rightarrow work\_saturday\}$.
Post-condition: $CS.\mathcal{A}$(Peter) = $CS.\mathcal{A}$(Peter) ∪ {manager, manager → work_saturday}.

Mary: I have nothing to say. *Say_nothing.*

John: I think you don't let me any choice ! Accept(manager, manager → work_saturday).
Pre-condition: The argument is acceptable.
Post-condition: $CS.\mathcal{A}$(John) = $CS.\mathcal{A}$(John) ∪ {manager, manager → work_saturday}.

Furthermore, the offer (V, J) is the most preferred one in X in the sense of definition 4.

In other words, all the agents have accepted the offer (V, J) and *Result(*$\mathcal{D}$*) = success.*

The negotiation dialogue ends with a compromise found by the agents to organize their meeting: in Valencia at the date J.

6 Conclusion

This paper has proposed a general formal framework for handling negotiation dialogues where autonomous agents aim at finding a common agreement about a collective choice. The agents are equipped with knowledge bases graded in certainty levels and gathering what they know about the environment, and with preference bases representing their more or less important goals.

The reasoning model of the agents is captured by a formal decision framework. The basic idea is that an agent utters and accepts offers which are supported by strong arguments. Similarly, agents refuse or challenge offers for which there exists at least one strong argument against them.

The interaction between agents is captured by a protocol which is run at most as many times as there non discussed offers, and such that at each run only one offer is discussed. If it is accepted by all the agents, then an agreement is found. In the opposite case, it is removed from the set of offers and another one is proposed.

In future work, we plan to propose a protocol less restrictive by considering stratified sets to store the rejected offers. A level of rejection will be computed to allow the affectation of the offers to the different sets. The last set in the stratification will gather the offers which are definitively rejected, i.e. those which are impossible. Once all the offers are studied without finding an acceptable one, the agents negotiate again on the set gathering the less rejected offers and proceed the same way. This requires that the agents revise their bases by being less demanding regarding their preferences.

References

1. L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proc. 4th Intl. Conf. on Multi-Agent Systems (ICMAS'00)*, pages 31–38, Boston, 2000.
2. L. Amgoud, N.Maudet, and S. Parsons. An argumentation-based semantics for agent communication languages. In *Proc. 15th Eur. Conf. on Artificial Intelligence (ECAI'02)*, pages 38–42, Lyon, 2002.
3. L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In *Proc. 14th Eur. Conf. on Artificial Intelligence (ECAI'00)*, pages 338–342, Berlin, 2000.
4. L. Amgoud and H. Prade. Reaching agreement through argumentation: A possibilistic approach. In *Proc. 9th Intl. Conf. on the Principles of Knowledge Representation and Reasoning (KR'04)*, pages 175–182, Whistler, 2004.
5. L. Amgoud and H. Prade. Using arguments for making decisions: A possibilistic logic approach. In *Proc. 20th Conf. on Uncertainty in Artificial Intelligence (UAI'04)*, pages 10–17, Banff, 2004.
6. S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda based framework for multi-issues negotiation. *Artificial Intelligence*, 152:1–45, 2004.
7. A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proc. 2nd Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'03)*, pages 883–890, Melbourne, 2003. ACM Press.
8. S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.
9. J. MacKenzie. Question-begging in non-cumulative systems. *Journal of philosophical logic*, 8:117–133, 1979.
10. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
11. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.
12. C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In *Proc. 4th Intl. Workshop on Agent Theories, Architectures and Languages (ATAL'97)*, pages 167–182, Providence, 1997.

Argumentation-Supported Information Distribution in a Multiagent System for Knowledge Management

Ramón F. Brena[1], Carlos I. Chesñevar[2], and José L. Aguirre[1]

[1] Centro de Sistemas Inteligentes – Tecnológico de Monterrey
64849 Monterrey, N.L., México
{ramon.brena, jlaguirre}@itesm.mx
[2] Artificial Intelligence Research Group – Department of Computer Science
Universitat de Lleida – Campus Cappont – C/Jaume II, 69 – E-25001 Lleida, Spain
cic@eps.udl.es

Abstract. Disseminating pieces of knowledge among the members of large organizations is a well known problem in Knowledge Management, involving several decision-making processes. The JITIK multiagent framework has been successfully used for just-in-time delivering highly customized notifications to the adequate users in large distributed organizations. However, in JITIK as well as in other similar approaches it is common to deal with incomplete information and conflicting policies, making difficult to make decisions about whether to deliver or not a specific piece of information or knowledge on the basis of a rationally justified procedure. This paper presents an approach to cope with this problem by integrating JITIK with a defeasible argumentation formalism. Conflicts among policies are solved on the basis of a dialectical analysis whose outcome determines whether a particular information item should be delivered to a specific user.

Keywords: Argumentation, knowledge management, information systems.

1 Introduction and Motivation

Information and Knowledge (IK) are each day more valuable assets in modern organizations [6, 12, 28]. Indeed, a central concern in Knowledge Management (KM) [29, 24] is to facilitate *knowledge flow*, either within an organization or from / to other relevant actors. IK distribution systems could be visualized as a kind of *information switch*, which finds adequate routing paths for IK from sources to consumers —the latter being normally humans and members of the given organization (employees, partners, etc.). IK is characterized by *metadata* (such as a content classification in terms of technical disciplines, intended audience, etc.) and users are characterized by *profiles*, which give the user function or position in the organization, rights and duties, interests, etc. Organizations typically have different criteria establishing their information distribution *policies*, and in many real situations these policies conflict with each other.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 279–296, 2006.

In the last years agent-based approaches have shown to be an interesting alternative to support information distribution systems. In particular, the JITIK[1] [8, 9, 1] multiagent framework has been successfully applied to just-in-time[2] disseminating pieces of IK among the members of large or distributed organizations. Clearly, in such organizations complex decision-making situations regarding IK distribution usually arise, specially in the presence of potentially incomplete information concerning metadata and user profiles, as well as competing policies, which may be complicated and could include several exceptions.

This paper presents a novel, argumentation-based approach to solve the problem defined above by integrating the JITIK platform with a defeasible argumentation formalism called Defeasible Logic Programming (DeLP) [21]. As a result, the resulting enhanced framework can efficiently solve IK-distribution problems involving conflicting policies among specific users, by applying a *dialectical analysis*. One important advantage of this argumentation-supported approach is that decisions concerning information distribution are fully explainable.

The structure of this paper is as follows: Section 2 presents the fundamentals of the JITIK system. Section 3 outlines the basics of DeLP. Section 4 describes the integration of JITIK and DeLP into an extended MAS framework. Section 5 presents a worked example. Section 6 outlines some implementation issues. Section 7 discusses related work, and finally Section 8 concludes.

2 The JITIK System

JITIK [8, 9, 1] is a multiagent-based system for disseminating pieces of IK among the members of a large or distributed organization, thus supporting a Knowledge-management function. It is aimed to deliver the right IK to the adequate people just-in-time. The JITIK agent model is shown in Fig. 1. *Personal Agents* work on behalf of the members of the organization. They filter and deliver useful content according to user preferences. The Site Agent provides of IK to the Personal Agents, acting as a broker between them and Service agents. *Service agents* collect and detect IK pieces that are supposed to be relevant for someone in the organization. Examples of service agents are the Web Service agents, which receive and process external requests, as well as monitor agents which are continuously monitoring sources of IK (web pages, databases, etc.). Other Service agents monitor at time intervals the state of an IK resource, like a web page, data in an enterprise's database, etc. The *Ontology agent* contains knowledge about the interest areas for the members of the organization and about its structure [10]. That knowledge is hierarchically described in the form of taxonomies, usually one for interest areas and one describing the structure of the organization. For example, in an academic institution, the interest areas could be the science domains in which the institution is specialized, and the organizational chart of the institution gives the structure of the organization. *Site agents*

[1] Just-In-Time Information and Knowledge.

[2] By "just-in-time" we mean that the right IK items are distributed at the right moment to the right people.

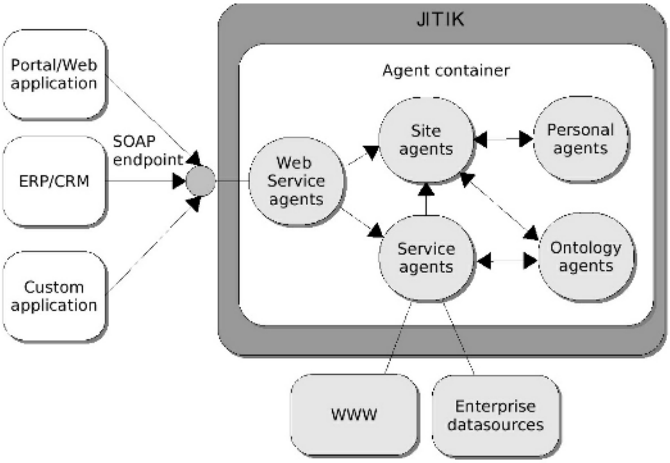


Fig. 1. The JITIK agent model

are the heart of a "cluster" composed by one site agent and several personal agents served by the former. In an organization, clusters would be associated to departments, divisions, etc., depending on the size of them. Networks can be made up connecting several site agents. Distributed organizations like multinational companies would have a web of many connected site agents. Among the services provided by JITIK we can mention the following:

- **Recommendation services:** A user's profile is represented by a set of points in the taxonomies, as each user could have many interests and could be located at different parts of the organizational structure. As JITIK keeps track of user interests and preferences it is able to recommend content to users on demand. Recommended content may be used in Portals or Web applications.
- **Subscription services:** JITIK allows users to subscribe to changes in specific areas. Also, users may customize the media and frequency of JITIK notifications using using simple web-based interfaces. Rules may be defined so as messages relative to certain topics are handled with higher priorities. A rule may state that several alerts should be sent to their cell-phone via SMS, and also define that interest-area messages be sent in a weekly summary via email. Organization managers may set high-level distribution rules.
- **Content distribution services:** Enterprise applications can deliver content to the system using its semantic-based content distribution services. When new content is received it is classified and distributed to those users who could be interested. Users receive the notifications of new content as specified by their own rules.

3 Defeasible Argumentation with DeLP

Logical models of defeasible argumentation [16, 38] have evolved in the last decade as a successful approach to formalize defeasible, commonsense reasoning. Recent research has shown that argumentation can be integrated in a growing number of real-world applications in a broad scope of areas such as legal reasoning [37], natural language processing [15], knowledge engineering [11], analysis of news reports [25] clustering [22], argumentation support systems [47], mediation systems and computer-supported collaborative argumentation [32, 41]. Over the last few years, argumentation has been gaining particular importance in the context of multi-agent systems [36, 42, 3, 43, 35, 4], providing tools for designing, implementing and analyzing sophisticated forms of interaction among rational agents.

Defeasible logic programming (DeLP) [21] is a particular general-purpose defeasible argumentation formalism based on logic programming. A defeasible logic program[3] is a set $K = (\Pi, \Delta)$ of Horn-like clauses, where Π and Δ stand for sets of strict and defeasible knowledge, respectively. The set Π of strict knowledge involves *strict rules* of the form $p \ \leftarrow \ q_1, \ldots, q_k$ and *facts* (strict rules with empty body), and it is assumed to be *non-contradictory.* The set Δ of defeasible knowledge involves *defeasible rules* of the form $p \mathrel{-\!\!\prec} \ q_1, \ldots, q_k$, which stands for "$q_1, \ldots q_k$ provide a *tentative reason* to believe p." The underlying logical language is that of extended logic programming, enriched with a special symbol " $\mathrel{-\!\!\prec}$ " to denote defeasible rules. Both default and classical negation are allowed (denoted not and $\sim$, resp.). Syntactically, the symbol " $\mathrel{-\!\!\prec}$ " is all that distinguishes a *defeasible* rule $p \mathrel{-\!\!\prec} \ q_1, \ldots q_k$ from a *strict* (non-defeasible) rule $p \ \leftarrow \ q_1, \ldots, q_k$. DeLP rules are thus Horn-like clauses to be thought of as *inference rules* rather than implications in the object language. Deriving literals in DeLP results in the construction of *arguments.*

Definition 1 (Argument). *Given a DeLP program $\mathcal{P}$, an* argument *$\mathcal{A}$ for a query q, denoted $\langle \mathcal{A}, q\rangle$, is a subset of ground instances of defeasible rules in $\mathcal{P}$ and a (possibly empty) set of default ground literals "*not *L", such that:*

1. *there exists a* defeasible derivation *for q from $\Pi \cup \mathcal{A}$;*
2. *$\Pi \cup \mathcal{A}$ is non-contradictory (i.e, $\Pi \cup \mathcal{A}$ does not entail two complementary literals p and $\sim p$ (or p and* not *p)), and*
3. *$\mathcal{A}$ is minimal with respect to set inclusion.*

An argument $\langle \mathcal{A}_1, Q_1\rangle$ is a sub-argument *of another argument $\langle \mathcal{A}_2, Q_2\rangle$ if $\mathcal{A}_1 \subseteq \mathcal{A}_2$. Given a DeLP program $\mathcal{P}$, $Args(\mathcal{P})$ denotes the set of all possible arguments that can be derived from $\mathcal{P}$.*

The notion of defeasible derivation corresponds to the usual query-driven SLD derivation used in logic programming, performed by backward chaining on both strict and defeasible rules; in this context a negated literal $\sim p$ is treated just as a new predicate name *no_p*. Minimality imposes a kind of 'Occam's razor

[3] When it is clear from the context we will simply refer to a defeasible logic program as a "DeLP program" or just "program".

principle' [44] on arguments. The non-contradiction requirement forbids the use of (ground instances of) defeasible rules in an argument $\mathcal{A}$ whenever $\Pi \cup \mathcal{A}$ entails two complementary literals.

Definition 2 (Counterargument – Defeat). *An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a* counterargument *for an argument $\langle \mathcal{A}_2, q_2 \rangle$ iff*

1. *There is an subargument $\langle \mathcal{A}, q \rangle$ of $\langle \mathcal{A}_2, q_2 \rangle$ such that the set $\Pi \cup \{q_1, q\}$ is contradictory.*
2. *A literal* not q_1 *is present in some rule in $\mathcal{A}_1$.*

A partial order $\preceq \subseteq Args(\mathcal{P}) \times Args(\mathcal{P})$ will be used as a preference criterion *among conflicting arguments. An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a* defeater *for an argument $\langle \mathcal{A}_2, q_2 \rangle$ if $\langle \mathcal{A}_1, q_1 \rangle$ counterargues $\langle \mathcal{A}_2, q_2 \rangle$, and $\langle \mathcal{A}_1, q_1 \rangle$ is preferred over $\langle \mathcal{A}_2, q_2 \rangle$ wrt $\preceq$. For cases (1) and (2) above, we distinguish between* proper *and* blocking defeaters *as follows:*

- *In case 1, the argument $\langle \mathcal{A}_1, q_1 \rangle$ will be called a* proper defeater *for $\langle \mathcal{A}_2, q_2 \rangle$ iff $\langle \mathcal{A}_1, q_1 \rangle$ is strictly preferred over $\langle \mathcal{A}, q \rangle$ wrt $\preceq$.*
- *In case 1, if $\langle \mathcal{A}_1, q_1 \rangle$ and $\langle \mathcal{A}, q \rangle$ are unrelated to each other, or in case 2, $\langle \mathcal{A}_1, q_1 \rangle$ will be called a* blocking defeater *for $\langle \mathcal{A}_2, q_2 \rangle$.*

Specificity [44] is used in DeLP as a syntax-based criterion among conflicting arguments, preferring those arguments which are *more informed* or *more direct* [44, 46]. However, other alternative partial orders could also be used.

An *argumentation line* starting in an argument $\langle \mathcal{A}_0, Q_0 \rangle$ (denoted $\lambda^{\langle \mathcal{A}_0, q_0 \rangle}$) is a sequence $[\langle \mathcal{A}_0, Q_0 \rangle, \langle \mathcal{A}_1, Q_1 \rangle, \langle \mathcal{A}_2, Q_2 \rangle, \ldots, \langle \mathcal{A}_n, Q_n \rangle \ldots]$ that can be thought of as an exchange of arguments between two parties, a *proponent* (evenly-indexed arguments) and an *opponent* (oddly-indexed arguments). Each $\langle \mathcal{A}_i, Q_i \rangle$ is a defeater for the previous argument $\langle \mathcal{A}_{i-1}, Q_{i-1} \rangle$ in the sequence, $i > 0$. In order to avoid *fallacious* reasoning, dialectics imposes additional constraints (viz. disallowing circular argumentation, enforcing the use of proper defeaters to defeat blocking defeaters, etc.[4]) on such an argument exchange to be considered rationally valid. An argumentation line satisfying the above restrictions is called *acceptable*, and can be proven to be finite [21]. Given a DeLP program $\mathcal{P}$ and an initial argument $\langle \mathcal{A}_0, Q_0 \rangle$, the set of all acceptable argumentation lines starting in $\langle \mathcal{A}_0, Q_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, Q_0 \rangle$ (ie., all possible dialogues rooted in $\langle \mathcal{A}_0, Q_0 \rangle$), formalized as a *dialectical tree.*

Definition 3 (Dialectical Tree). *A* dialectical tree *for an argument $\langle \mathcal{A}_0, Q_0 \rangle$, denoted $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$, is a tree structure defined as follows:*

1. *The root node of $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ is $\langle \mathcal{A}_0, Q_0 \rangle$.*
2. *$\langle \mathcal{B}', H' \rangle$ is an immediate children of $\langle \mathcal{B}, H \rangle$ iff there exists an acceptable argumentation line $\lambda^{\langle \mathcal{A}_0, Q_0 \rangle} = [\langle \mathcal{A}_0, Q_0 \rangle, \langle \mathcal{A}_1, Q_1 \rangle, \ldots, \langle \mathcal{A}_n, Q_n \rangle]$ such that there are two elements $\langle \mathcal{A}_{i+1}, Q_{i+1} \rangle = \langle \mathcal{B}', H' \rangle$ and $\langle \mathcal{A}_i, Q_i \rangle = \langle \mathcal{B}, H \rangle$, for some $i = 0 \ldots n-1$.*

[4] For an in-depth treatment of dialectical constraints in DeLP the reader is referred to [21].

Nodes in a dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ can be marked as *undefeated* and *defeated* nodes (U-nodes and D-nodes, resp.). A dialectical tree will be marked as an AND-OR tree: all leaves in $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ will be marked U-nodes (as they have no defeaters), and every inner node is to be marked as *D-node* iff it has at least one U-node as a child, and as *U-node* otherwise. An argument $\langle \mathcal{A}_0, Q_0 \rangle$ is ultimately accepted as valid (or *warranted*) wrt a DeLP program $\mathcal{P}$ iff the root of its associated dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ is labeled as *U-node.*

Given a DeLP program $\mathcal{P}$, solving a query q wrt $\mathcal{P}$ accounts for determining whether q is supported by a warranted argument. Different doxastic attitudes are distinguished when answering q according to the associated status of warrant, in particular:

1. Believe q (resp. $\sim q$) when there is a warranted argument for q (resp. $\sim q$) that follows from $\mathcal{P}$;
2. Believe q is *undecided* whenever neither q nor $\sim q$ are supported by warranted arguments in $\mathcal{P}$.

It should be noted that that the computation of warrant cannot lead to contradiction [21]: if there exists a warranted argument $\langle A, h \rangle$ on the basis of a program $\mathcal{P}$, then there is no warranted argument $\langle B, \sim h \rangle$ based on $\mathcal{P}$.

4 Integrating JITIK with DeLP

The JITIK framework, as it stands, can take into consideration hierarchies for users and content classification for determining how distribution rules are to be applied. In the case of policies with exceptions, or competing policies, specialized criteria have to be explicitly encoded in both Site and Personal agents. In many respects such an approach is undesirable. On the one hand, such changes involve modifying the underlying decision algorithm. The correctness of such changes may be difficult to test, as unexpected side-effects might arise for new future cases. On the other hand, the knowledge engineer should be able to encode knowledge as declaratively as possible, including the possibility of representing competing policies. Such knowledge should be independent of the rational procedure for determining which is the winning policy when conflicting situations arise.

Our proposal consists of integrating the JITIK framework with DeLP, incorporating distribution policies for Site Agents explicitly in terms for defeasible logic programs. As explained in Section 2, a JITIK Site Agent Ag_S is responsible for distributing IK among different Personal Agents $Ag_1, \ldots Ag_n$. We will use DeLP programs to represent the knowledge of these agents. Thus, preferences of different Personal Agents $Ag_1, \ldots, Ag_n$ will be represented as DeLP programs $\mathcal{P}_{Ag_1}, \ldots, \mathcal{P}_{Ag_n}$. Distribution policies and preferences of the Site Agent Ag_S will be represented by another DeLP program $\mathcal{P}_S$. In contrast with the programs associated with Personal Agents, this program $\mathcal{P}_S$ will contain corporate rules defining hierarchies and (possibly conflicting) policies for IK distribution among personal agents.

Given a list $L = [Item_1, \ldots, Item_i]$ of IK items to be distributed by the Site Agent Ag_S among different Personal Agents $Ag_1, \ldots, Ag_n$, a distinguished

```
ALGORITHM DistributeItems
{Executed by Site Agent $Ag_S$ to decide distribution of items in $L$}
INPUT: List $L = [item_1, \ldots, item_k]$ of incoming items
       DeLP program $\mathcal{P}_S$ for Site Agent $Ag_S$
       DeLP programs $\mathcal{P}_1, \ldots, \mathcal{P}_n$ for Personal Agents depending from $Ag_S$
OUTPUT: Item distribution to Personal Agents
        according to policies and user preferences
BEGIN
  $\mathcal{P}'_S := \mathcal{P}_S \cup \{info(item_1), \ldots, info(item_k)\}$
  {Encode incoming items as new facts for Site Agent}
  FOR every item $I \in L$
    FOR every Personal Agent $Ag_i$ supervised by $Ag_S$
      Let $\mathcal{P} = \mathcal{P}'_S \cup \mathcal{P}_{Ag_i}$
      Using program $\mathcal{P}$, solve query $distribute(Item, Ag_i)$
      IF $distribute(Item, Ag_i)$ is warranted
        THEN
        Send message $I$ to agent $Ag_i$
END
```

Fig. 2. Algorithm for Knowledge Distribution using DeLP in a JITIK Site Agent

predicate $distribute(I, User)$ will be used to determine whether a particular IK item $I \in L$ is intended to be delivered to a specific user $User$. The above query will be solved by the DeLP inference engine on the basis of a program $\mathcal{P}$ which will take into account the Site Agent's knowledge, the metadata corresponding to the incoming items to be distributed and the personal preferences of the different users involved. This is made explicit in algorithm shown in Fig. 2. Solving queries based on the *distribute* predicate wrt the DeLP inference engine will automate the decision making process for Site Agents, providing a rationally justified decision even for very complex cases, as we will see in the next section.

5 A Worked Example

In this section we present an illustrative example of how DeLP is integrated into the JITIK system to distribute IK items to users. We assume a typical corporate environment where people could have different rights and responsibilities (CEO, managers, supervisors, etc.). These people (users) will belong to different areas of the organization (production, marketing, etc.), and will have different personal interests and preferences which are characterized by their corresponding Personal Agents. In our example IK items will correspond to memos, which have to be delivered by a Site Agent to different users according to the organization policies. Personal interests and preferences of the different users involved should be also taken into account.

Within our organization, areas and topics are organized in *hierarchies*. Thus, for example, a hierarchy of topics for memos could be "computers – hardware – processors". The Site Agent is required to take this into account, performing inheritance reasoning to infer consequences related to subareas: if a user is not interested in memos related to hardware, he will not be interested in memos related to processors either. Note that other organization policies could add *exceptions* to such hierarchies, e.g. by stipulating that a certain memo is mandatory, and should be delivered without regarding the user preferences.

In our example, IK items made available from the organization to the Site Agent will correspond to different memos, which will be encoded with a predicate $info(Id, A, L, M, T, S)$, meaning that the memo with unique identifier Id is about area A and it can be accessed by users of at least level L. Other attributes associated with the memo are whether it is mandatory ($M = 1$) or optional ($M = 0$), top secret ($T = 1$) or not ($T = 0$) and is originated at source S. Thus, the fact

$$\mathit{info}(\mathit{id}_3, \mathit{computers}, \mathit{manager}, 0, 0, \mathit{marketing}) \leftarrow$$

indicates that the memo id_3 is about *computers*, it is intended at least for managers, it is not mandatory nor secret, and it has been produced by the department of marketing.

5.1 Characterizing Organization knowledge in Site and Personal Agents

Figure 3 shows a sample DeLP code associated with a Site and a Personal agent in our organization.[5] Strict rules s_1 to s_9 characterize permissions and extract information from memos. Rule s_1 defines that a user P is *allowed* access to item I if he/she has the required *permissions.* Granted permissions are given as facts (f_1, f_2 and f_3). Permissions are also propagated using the strict rules s_4, s_5 and s_6, where the binary predicate *depends* establishes the organization hierarchy, stating that the first argument person is (transitively) subordinated to the second one. This predicate is calculated as the transitive closure of a basic predicate *subordinate* (defined by facts f_4 and f_5), which establishes subordinate relationships pairwise. Thus, having e.g. granted permissions as CEO allows the CEO to have access to every memo corresponding to lower level permissions. Note that the predicate *subordinate* uses generic *roles* as arguments, not specific person identifiers. Rule s_2 and s_3 define the predicate $isAbout(I, A)$ as an information hierarchy among subfields. The basic case corresponds to a subfield for which specific information is available (rule s_2). Note that in our particular example facts f_6 and f_7 define the basic relationships in this hierarchy. Finally, rules s_7, s_8 and s_9 define auxiliary predicates *source*, *mandatory* (yes/no) and *topsecret* (yes/no) which allow to extract this particular attributes from the memos to be distributed that just extract information from *info*, facts, simplifying the subsequent analysis.

Let us now consider the defeasible rules for our Site Agent. Rule d_1 defines when an item I is usually of interest for a specific user U, on the basis of the user's personal preferences. Rule d_2 and d_4 define a policy for memo distribution

[5] Note that we distinguish strict rules, defeasible rules, and facts by using s_i, d_i and f_i as clause identifiers, respectively.

Site Agent Knowledge

Strict rules

s_1) $allowed(I,U) \leftarrow info(I,A,L,M,T,S),\ permissions(U,L).$

s_2) $isAbout(I,A) \leftarrow info(I,A,L,M,T,S)$

s_3) $isAbout(I,A) \leftarrow subField(SuperA,A),\ isAbout(I,SuperA).$

s_4) $permissions(U,X) \leftarrow depends(X,Y),\ permissions(U,Y).$

s_5) $depends(X,Y) \leftarrow subordinate(X,Y).$

s_6) $depends(X,Z) \leftarrow subordinate(Y,Z),\ depends(X,Y).$

s_7) $source(I,S) \leftarrow info(I,\_,\_,\_,\_,S).$

s_8) $mandatory(I) \leftarrow info(I,\_,\_,1,\_,\_).$

s_9) $topsecret(I) \leftarrow info(I,\_,\_,\_,1,\_).$

Defeasible rules

d_1) $interest(I,U) \mathrel{-\!\!\prec} isAbout(I,A),\ interestField(A,U).$

d_2) $distribute(I,U) \mathrel{-\!\!\prec} allowed(I,U),\ mandatory(I,U).$

d_3) $\sim mandatory(I,U) \mathrel{-\!\!\prec} permissions(U,manager),\ \sim interest(I,U),$
$\mathsf{not}\ topsecret(I).$

d_4) $distribute(I,U) \mathrel{-\!\!\prec} allowed(I,U),\ interest(I,U).$

Facts

Granted Permissions within the organization

f_1) $permissions(joe,manager) \leftarrow$

f_2) $permissions(peter,everybody) \leftarrow$

f_3) $permissions(dana,ceo) \leftarrow$

People Hierarchy

f_4) $subordinate(everybody,manager) \leftarrow$

f_5) $subordinate(manager,ceo) \leftarrow$

Field Hierarchy

f_6) $subField(hardware,computers) \leftarrow$

f_7) $subField(processors,hardware) \leftarrow$

Information Items as facts

f_8) $info(id_1,computers,everybody,0,0,external) \leftarrow$

f_9) $info(id_2,computers,everybody,0,0,techdept) \leftarrow$

f_{10}) $info(id_5,processors,manager,1,1,techdept) \leftarrow$

Personal Agent Knowledge

Defeasible rules

d'_1) $\sim interest(I,joe) \mathrel{-\!\!\prec} isAbout(I,A),\ interestField(A,joe).$
$source(I,S),\ \sim relies(joe,S).$

d'_2) $\sim interest(I,joe) \mathrel{-\!\!\prec} isAbout(I,A),\ interestField(A,joe),$
$isAbout(I,SuperA),\ \sim interestField(SuperA,joe).$

Facts

User Preferences

f'_1) $interestField(computers,joe) \leftarrow$

f'_2) $\sim interestField(hardware,joe) \leftarrow$

f'_3) $relies(joe,techdept) \leftarrow$

f'_4) $\sim relies(joe,external) \leftarrow$

Fig. 3. DeLP code for a Site Agent and a Personal Agent in JITIK

in our organization: a) an item (memo) I should be delivered to a user U if he is allowed to read this memo, and it is mandatory for him to read it; b) an item I should be delivered to a user U if he is allowed to read it, and it is interesting for him. Rule d_3 provides an exception for mandatory memos: users which have at least permission as managers are not obliged to read memos they are not interested in, *unless* they are top secret ones.[6]

Finally, let us consider the DeLP program associated with a particular Personal Agent (e.g. Joe). A number of facts represent Joe's preferences: which are his interest fields, and his personal belief about other parts of the organization (e.g. reliability with respect to the source of incoming memo).[7] Joe can provide also a number of defeasible rules associated with his preferences. Rule d'_1 establishes that Joe is not interested in a memo coming from an unreliable source. Rule d'_2 defines how to handle "negative inheritance" within the hierarchy of interests: Joe is not interested in any area A which is a subarea of another area $SuperA$, such that $SuperA$ is not interesting for him (e.g. if he is interested in computers but not interested in hardware, he will not be interested for a memo about processors, as processors are a subarea of hardware).

5.2 Solving Conflicts for Information Distribution as DeLP Queries

Let us assume that there is a list of information items $[Memo_1, Memo_2, Memo_5]$ corresponding to memos to be distributed by our Site Agent, which encodes organization policies as a DeLP program $\mathcal{P}_S$. By applying the algorithm given in Fig. 2, these items will be encoded temporarily as a set $\mathcal{P}_{items} = \{info(Memo_1),$ $info(Memo_2)$, $info(Memo_5)\}$ (see Fig. 3).

For the sake of simplicity, we will assume that there is only one single Personal Agent involved, associated with a specific user *joe*, whose role is *manager*. Joe's Personal Agent mirrors his preferences in terms of a DeLP program $\mathcal{P}_{joe} =$ $\{d'_1, d'_2, f'_1, f'_2, f'_3, f'_4\}$, which together with $\mathcal{P}_S$ and $\mathcal{P}_{items}$ will provide the knowledge necessary to decide which IK items should be delivered to this specific user. Following the algorithm in Fig. 2, the Site Agent will have to solve the queries $distribute(id_1, joe)$, $distribute(id_2, joe)$ and $distribute(id_5, joe)$ wrt the DeLP program $\mathcal{P}_S \cup \mathcal{P}_{items} \cup \mathcal{P}_{joe}$. We will show next how every one of these queries is solved in different examples that show how DeLP deals with conflicts among organization policies and user preferences.

Example 1. Consider the query $distribute(id_1, joe)$. In this case the DeLP inference engine will find the argument $\langle \mathcal{A}_1, distribute(id_1, joe)\rangle$, with[8] $\mathcal{A}_1 =$

$$\begin{aligned}\{distribute(id_1, joe) &\prec allowed(id_1, joe),\\ & interest(id_1, joe);\\ interest(id_1, joe) &\prec isAbout(id_1, computers),\\ & interestField(computers, joe)\}\end{aligned}$$

[6] Note how this last condition is expressed in terms of default negation not in rule d_3.

[7] Note the use of explicit negation in these predicates.

[8] For the sake of clarity, we use semicolons to separate elements in an argument $\mathcal{A} =$ $\{e_1\ ;\ e_2\ ;\ \ldots;\ e_k\ \}$.

However, in this case, a defeater $\langle \mathcal{A}_2, \sim interest(id_1, joe)\rangle$ for the argument $\langle \mathcal{A}_1, distribute(id_1, joe)\rangle$ will be found, with $\mathcal{A}_2 =$

$$\begin{aligned}\{\sim interest(id_1, joe) \;\prec\; & isAbout(id_1, computers)\\ & interestField(computers, joe)\\ & source(id_1, external),\\ & \sim relies(joe, external).\}\end{aligned}$$

Note that in this case, id_1 comes from an external source, and according to *joe*'s preference criteria, external sources are unreliable. Hence the Site Agent will not deliver this information item to him. In this case, the dialectical tree $\mathcal{T}_{\langle \mathcal{A}_1, distribute(id_1, joe)\rangle}$ has two nodes in a single branch (see Figure 4-i). There are no other arguments to consider, and $\langle \mathcal{A}_1, distribute(id_1, joe)\rangle$ is not warranted.

Example 2. Consider now the query $distribute(id_2, joe)$. There is an argument $\langle \mathcal{B}_1, distribute(id_1, joe)\rangle$, with $\mathcal{B}_1 =$

$$\begin{aligned}\{distribute(id_2, joe) \;\prec\; & allowed(id_2, joe),\\ & interest(id_2, joe);\\ interest(id_2, joe) \;\prec\; & isAbout(id_2, computers),\\ & interestField(computers, joe)\}\end{aligned}$$

This argument has no defeaters. Hence the dialectical tree $\mathcal{T}_{\langle \mathcal{B}_1, distribute(id_2, joe)\rangle}$ has a single node, marked as U-node (see Fig. 4-ii). The original argument is therefore warranted.

Example 3. Finally consider the query $distribute(id_5, joe)$. There is an argument $\langle \mathcal{C}_1, distribute(id_1, joe)\rangle$, with $\mathcal{C}_1 =$

$$\begin{aligned}\{distribute(id_5, joe) \;\prec\; & allowed(id_5, joe), interest(id_5, joe);\\ interest(id_5, joe) \;\prec\; & isAbout(id_5, computers),\\ & interestField(computers, joe)\}\end{aligned}$$

However, in this case, a defeater $\langle \mathcal{C}_2, \sim interest(id_5, joe)\rangle$ for the argument $\langle \mathcal{C}_1, distribute(id_5, joe)\rangle$ can be found, with $\mathcal{C}_2 =$

$$\begin{aligned}\{\sim interest(id_5, joe) \;\prec\; & isAbout(id_5, computers),\\ & interestField(computers, joe),\\ & isAbout(id_5, hardware),\\ & \sim interestField(hardware, joe).\}\end{aligned}$$

As in Example 1, the argument $\langle \mathcal{C}_1, distribute(id_1, joe)\rangle$ is not warranted (see Fig. 4-iii). The DeLP inference engine searches then for alternative arguments for $distribute(id_5, joe)$. There is another one, namely $\langle \mathcal{D}_1, distribute(id_5, joe)\rangle$, with $\mathcal{D}_1$=

$$\{distribute(id_2, joe) \prec allowed(id_2, joe), mandatory(id_5, joe)\}$$

$\langle \mathcal{A}_1, distribute(id_1, joe)\rangle^D$ $\vert$ $\langle \mathcal{A}_2, \sim interest(id_1, joe)\rangle^U$ (i)	$\langle \mathcal{B}_1, distribute(id_2, joe)\rangle^U$ (ii)
$\langle \mathcal{C}_1, distribute(id_5, joe)\rangle^D$ $\vert$ $\langle \mathcal{C}_2, \sim interest(id_5, joe)\rangle^U$ (iii)	$\langle \mathcal{D}_1, distribute(id_5, joe)\rangle^U$ $\vert$ $\langle \mathcal{D}_2, \sim mandatory(id_5, joe)\rangle^D$ $\vert$ $\langle \emptyset, topsecret(id_5)\rangle^U$ (iv)

Fig. 4. Dialectical trees for queries $distribute(id_1, joe)$, $distribute(id_2, joe)$ and $distribute(id_5, joe)$ (examples 1,2 and 3)

which in this case is defeated by another argument $\langle \mathcal{D}_2, \sim mandatory(id_5, joe)\rangle$, with $\mathcal{D}_2 =$

$$\begin{aligned}
\{\sim mandatory(id_5, joe) &\mathrel{-\!\!\prec} permissions(joe, manager),\\
&\quad\ \sim interest(id_5, joe),\\
&\quad\ \textsf{not}\ topsecret(id_5);\\
\sim interest(id_5, joe) &\mathrel{-\!\!\prec} isAbout(id_5, computers),\\
&\quad\ interestField(computers, joe),\\
&\quad\ isAbout(id_5, hardware),\\
&\quad\ \sim interestField(hardware, joe)\}
\end{aligned}$$

which is on its turn defeated by a third, empty argument $\langle \mathcal{D}_3, topsecret(id_5)\rangle$, with $\mathcal{D}_3 = \emptyset$ (note that $topsecret(id_5)$ is logically entailed by the strict knowledge of the Site Agent, and hence no defeasible information is needed). Argument $\langle \mathcal{D}_3, topsecret(id_5)\rangle$ defeats $\langle \mathcal{D}_2, \sim mandatory(id_5, joe)\rangle$, reinstating the argument $\langle \mathcal{D}_1, distribute(id_5, joe)\rangle$. Note that in this particular case the argument $\langle \mathcal{D}_1, distribute(id_5, joe)\rangle$ is warranted (see Fig. 4-iv).

After solving the different queries as shown in the previous examples, the Site Agent will proceed to deliver only memos id_2 and id_5 to joe's Personal Agent, but not memo id_1.

6 Implementation Issues

An implementation of JITIK that contains the Site Agent, the Personal Agents, an Ontology Agent and various Service Agents (web monitoring and others) has been reported in [40, 13], using the Jade [7] agent platform and the Java programming language.

As discussed in Section 3, solving queries in DeLP is not an easy task, and as such it requires an efficient computational implementation [20]. To this end, a particular abstract machine called JAM (Justification Abstract Machine) has been developed [21]. The JAM provides an argument-based extension of the traditional Warren abstract machine for Prolog [2]. A full-fledged implementation of DeLP based on this abstract machine is freely available online[9], including facilities for visualizing arguments and dialectical trees. On the basis of this abstract machine a Java-based integrated development environment was also implemented, which allows not only on-line compilation and query solving of DeLP code but also visualization of dialectical trees using a graphic interface [45]. This particular DeLP implementation has been successfully applied in a number of real-world applications such as web recommendation systems [17], clustering classification [22], natural language processing [15, 18] and web personalization [23].

Argument-based frameworks have shown to be particularly suited as an alternative to traditional non-monotonic logics for modelling commonsense reasoning (for discussion see in [16, 38]). We think that an argument-based framework like DeLP is particularly well suited for solving knowledge management problems like the ones presented in this paper. On the one hand, declarative knowledge can be encoded using logic programming style, which provides a natural and powerful rule-based language. On the other hand, the underlying inference procedure for DeLP extends naturally the traditional logic programming model, using backward chaining on arguments to compute dialectical trees. Although inference involves different aspects, the procedure is defined modularly: arguments are compared as structures on the basis of a comparison criterion which could be suitably modified or extended to consider additional elements (e.g. prioritizing defeasible rules, and preferring those arguments using rules with higher priority). Thus, a DeLP programmer does not need to encode exceptions at rule level explicitly, in contrast with other approaches (e.g. [5]). Additional aspects like incorporating vague knowledge and possibilistic reasoning into DeLP have also been explored recently, resulting in an extended framework called P-DeLP [19].

We must remark that our experiments regarding this JITIK-DeLP integration only account currently as a "proof of concept" prototype, as we have not been able yet to carry out thorough evaluations in the context of a real-world application. In particular, the sample problem presented in Section 5 was encoded and solved using the mentioned Java-based DeLP environment, from which only the part in charge of query solving was used.

In our prototype implementation the DeLP module loads at startup the rule and data bases from both the Site Agent and the Personal Agents, merging them in a single DeLP program. When the Site Agent receives a notification of a Service Agent it invokes the DeLP service module, passing the fact representing the notified IK and the query asking to whom the information has to be distributed. The DeLP module makes the argumentation process based on the query received

[9] See `http://lidia.cs.uns.edu.ar/DeLP`

and gives the results to the Site Agent, which sends a notification message to the corresponding Personal Agents.

7 Related Work and Discussion

To the best of our knowledge there are virtually no other works in the area of argumentation-based automated information distribution. In [31] the authors present a defeasible reasoning method for dealing with workflow processes, and it shares some goals with our work, as they also are able to deal with exceptions and imprecise information, but the application domain is quite different. A somehow related research is reported in [30] about methods for helping in decision-making processes using argumentation. Besides the differences in the intended application of this system there is also an important difference in the approach as they use static predefined argumentation schemas, whereas here we propose a general method for constructing arguments that is not restricted to a finite number of argument structures. Other works related to ours involve decision making and negotiation using argumentation among agents [34, 36, 39]. In contrast, in our system the argumentation process itself is not distributed, and it always takes place in the DeLP inference engine called by the Site Agent. Other argumentation-based decision-support proposals focus on the planning process for workgroup support, like the "dialectical planning" of Karacapilidis [27]. In none of the above references the problem of IK distribution is considered as done in this paper.

An interesting research issue is to study how the JITIK-DeLP model could be applied for characterizing *institutionalised power* [26]. It is a standard features in norm-governed organisations that particular agents (usually when acting in specific roles) be empowered to create specified kinds of states of affairs (which have conventional significance or meaning inside the institution in question, though not necessarily outside it). The power to create an institutional fact is in general distinguished from the permission to exercise that power. Thus, for example, a central concern in Virtual Organisations (VO) management [48, 33] is to monitor information flow among members wishing to cooperate on a shared project across organisational boundaries. Confidential issues (know-how and trade secrets) are a central concern. Therefore, a VO member (employed by a given organisation A) might be *empowered* to authorize access to A's trade secret, without being *permitted* to do so (if, e.g., she/he has signed a non-disclosure agreement). The notion of permissions presented in this paper could be thus extended to include more complex elements from the theory and practice of VO management.

8 Conclusions and Future Work

We have presented a novel argument-based approach for supporting IK distribution processes in large organizations by providing an integration of the JITIK multiagent platform with a defeasible argumentation formalism. As we have shown in this paper, the main advantage obtained through the use of an argumentation engine in JITIK is an increased flexibility, as it is not necessary to explicitly encode actions for every possible situation. This is particularly important in corporate environments with potentially conflicting IK distribution criteria.

Our approach is applicable in general to the distribution of IK that can be characterized by symbolic metadata expressed as ground terms in predicate logic. A Site Agent would use a DeLP program to represent corporate rules and organization IK, whereas their associated Personal Agents would use other user-defined DeLP programs for characterizing user profiles. In practice, end-users should not be allowed to establish arbitrary rules, and several ways for enforcing restrictions are possible (e.g. by providing ontology-based rule editors, conveniently set up for different kinds of organization users).

In the near future we also intend to extend our prototype into larger applications in real-world environments. An interesting challenge is the development of a distributed version of DeLP that could allow several JITIK Site Agents to perform collaborative decision making.

Acknowledgments

The authors would like to thank anonymous reviewers for their suggestions to improve the original version of this paper. This work was supported by the Monterrey Tech CAT-011 research chair, by Projects TIC2003-00950, TIN 2004-07933-C03-03, by Ramón y Cajal Program (MCyT, Spain) and by CONICET (Argentina).

References

1. J.L. Aguirre, R. Brena, and F.J. Cantu. Multiagent-based knowledge networks. *Expert Systems with Applications*, 20(1):65–75, Jan 2001.
2. H. Ait-Kaci. *Warren's Abstract Machine: A Tutorial Reconstruction.* The MIT Press, 1991.
3. L. Amgoud, N. Maudet, and S. Parsons. An argumentation-based semantics for agent communication languages. In *Proc. of the 15th. European Conference in Artificial (ECAI), Lyon, France*, pages 38–42, 2002.
4. L. Amgoud and H. Prade. Reaching agreement through argumentation: A possibilistic approach. In *Proc. of 9th Intl. Conf. on Knowledge Representation and Reasoning (KR 2004)*, pages 175–182, 2004.
5. G. Antoniou, D. Billington, G. Governatori, and M. Maher. Representation results for defeasible logic. *ACM Trans. on Computational Logic*, 2(2):255–287, 2001.
6. R. Atkinson, R. Court, and J. Ward. The knowledge economy: Knowledge producers and knowledge users. *The New Economic Index*, page http://www.neweconomyindex.org/, November 1998.
7. F. Bellifemine, A. Poggi, and G. Rimassa. Jade - a fipa-compliant agent framework. In *Proceedings of PAAM99, London*, 1999.

8. R. Brena, J. L. Aguirre, and A. C. Trevino. Just-in-time information and knowledge: Agent technology for km bussiness process. In *Proc. of the 2001 IEEE Conference on Systems, Man and Cybernetics, Tucson, Arizona*. IEEE Press, 2001.
9. R. Brena, J. L. Aguirre, and A. C. Trevino. Just-in-time knowledge flow for distributed organizations using agents technology. In *Proc. of the 2001 Knowledge Technologies 2001 Conf., Austin, Texas, 4-7 March 2001*, 2001.
10. R. Brena and H. Ceballos. A hybrid local-global approach for handling ontologies in a multiagent system. In Ronald R. Yager and Vassil S. Sgurev, editors, *Proc. of the 2004 Intelligent Systems International Conference, Varna, Bulgaria*, pages 261–266. IEEE Press, 2004.
11. D. Carbogim, D. Robertson, and J. Lee. Argument-based applications to knowledge engineering. *The Knowledge Engineering Review*, 15(2):119–149, 2000.
12. J. Carrillo. Managing knowledge-based value systems. *Journal of Knowledge Management*, 1(4), June 1998.
13. H. Ceballos and R. Brena. Finding compromises between local and global ontology querying in multiagent systems. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences Procs.*, volume 3291 of *LNCS*, pages 999–1011. Springer Verlag, 2004.
14. C. Chesñevar, J. Dix, F. Stolzenburg, and G. Simari. Relating Defeasible and Normal Logic Programming through Transformation Properties. *Theoretical Computer Science*, 290(1):499–529, 2003.
15. C. Chesñevar and A. Maguitman. An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus. In *Proc. of the ECAI-2004 Conference. Valencia, Spain*, pages 581–585, August 2004.
16. C. Chesñevar, A. Maguitman, and R. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.
17. C. Chesñevar, A. Maguitman, and G. Simari. Argument-Based Critics and Recommenders: A Qualitative Perspective on User Support Systems. *Journal of Data and Knowledge Engineering (to appear)*, 2005.
18. C. Chesñevar, M.Sabate, and A. Maguitman. An argument-based decision support system for assessing natural language usage on the basis of the web corpus. *Intl. Journal of Intelligent Systems (to appear)*, 2005.
19. C. Chesñevar, G. Simari, T. Alsinet, and Lluis Godo. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. Intl. Conf. in Uncertainty in Artificial Intelligence (UAI), Canada*, pages 76–84, July 2004.
20. C. Chesñevar, G. Simari, and L. Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *Proc. of the Intl. 8th LPNMR Conference. Cosenza, Univ. of Calabria, Italy. Published in LNAI Springer Series, Vol. 3662*, pages 158–171. Springer, September 2005.
21. A. García and G. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
22. S. Gómez and C. Chesñevar. A Hybrid Approach to Pattern Classification Using Neural Networks and Defeasible Argumentation. In *Proc. of 17th Intl. FLAIRS Conf. Miami, Florida, USA*, pages 393–398. AAAI Press, May 2004.
23. S. Gómez, C. Chesñevar, and G. Simari. Incorporating defeasible knowledge and argumentative reasoning in web-based forms. In *Proc. of the 3rd Intl. Workshop on Intelligent Techniques for Web Personalization (ITWP). 19th IJCAI Conference, Edimburgh, UK (to appear)*, August 2005.
24. F. Horibe. *Managing Knowledge Workers.* John Wiley and Sons, 1999.

25. A. Hunter. Hybrid argumentation systems for structured news reports. *Knowledge Engineering Review*, (16):295–329, 2001.
26. A. Jones and M. Sergot. A formal characterisation of institutionalised power. *J. of the IGLP*, 4(3):429–445, 1996.
27. N. Karacapilidis and T. Gordon. Dialectical planning: Designing a mediating system for group decision making, 1996.
28. J. Liebowitz and T. Beckman. *Knowledge Organizations.* St. Lucie Press, 1998.
29. J. Liebowitz and L. Wilcox. *Knowledge Management.* CRC Press, 1997.
30. J. Lowrance, I. Harrison, and A. Rodriguez. Structured argumentation for analysis. In *Procs. of the 12th Intl. Conf. on Systems Research, Informatics, and Cybernetics*, pages 47–57, Baden-Baden, Germany, Aug 2000.
31. Z. Luo, A. Sheth, J. Miller, and K. Kochut. Defeasible workflow, its computation and exception handling. In *Proc. of the CSCW-98 Workshop: Towards Adaptive Workflow Systems, Seattle, WA*, November 1998.
32. N. Maudet and D. J. Moore. Dialogue games for computer supported collaborative argumentation. In *Proceedings of the 1st Workshop on Computer supported collaborative argumentation (CSCA99)*, Standford, USA, 1999.
33. T. Norman, A. Preece, S. Chalmers, N. Jennings, M. Luck, V. Dang, T. Nguyen, V. Deora, J. Shao, W. Gray, and N. Fiddian. Conoise: Agent-based formation of virtual organisations. In *Research and Development in Intelligent Systems XX: Proceedings of AI2003, 23rd International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 353–366, 2003.
34. S. Parsons and N. R. Jennings. Argumentation and multi-agent decision making. In *Proceedings of the AAAI Spring Symposium on Interactive and Mixed-Initiative Decision Making*, pages 89–91, Stanford, USA, 1998.
35. S. Parsons and P. McBurney. Argumentation-based Dialogues for Agent Coordination. *Group Decision and Negotiation (to appear)*, 2004.
36. S. Parsons, C. Sierrra, and N. Jennings. Agents that Reason and Negotiate by Arguing. *Journal of Logic and Computation*, 8:261–292, 1998.
37. H. Prakken and G. Sartor. The role of logic in computational models of legal argument - a critical survey. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond*, pages 342–380. Springer, 2002.
38. H. Prakken and G. Vreeswijk. Logical Systems for Defeasible Argumentation. In D. Gabbay and F.Guenther, editors, *Handbook of Phil. Logic*, pages 219–318. Kluwer, 2002.
39. I. Rahwan, S. Ramchurn, N. Jennings, P. Mcburney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowl. Eng. Rev.*, 18(4):343–375, 2003.
40. E. Ramirez and R. Brena. Web-enabling multiagent systems. In *Advances in Artificial Intelligence – IBERAMIA 2004: 9th Ibero-American Conference on AI*, volume 3315 of *LNCS*, pages 53–61. Springer Verlag, 2004.
41. C. Reed and D. Walton. Applications of argumentation schemes. In *Conference of the Ontario Society for the Study of Argument (OSSA2001), Windsor, Canada*, 2001.
42. F. Sadri, F. Toni, and P. Torroni. Dialogues for negotiation: Agent varieties and dialogue sequences. In *Proc. of 8th Intl. Workshop on Agent Theories, Architectures and Languages (ATAL 2001)*, pages 405–421, 2001.
43. C. Sierra and P. Noriega. Agent-mediated interaction. from auctions to negotiation and argumentation. In *Foundations and Applications of Multi-Agent Systems – In LNCS Series, Vol. 2403*, pages 27–48. Springer, 2002.
44. G. Simari and R. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Art. Intelligence*, 53:125–157, 1992.

45. A. Stankevicius, A. Garcia, and G. Simari. Compilation techniques for defeasible logic programs. In *Proc. of the 6th Intl. Congress on Informatics Engineering*, pages 1530–1541. Univ. de Buenos Aires, Bs. Aires, Argentina, Ed. Fiuba, April 2000.
46. F. Stolzenburg, A. García, C. Chesñevar, and G. Simari. Computing Generalized Specificity. *J. of Non-Classical Logics*, 13(1):87–113, 2003.
47. B. Verheij. Artificial argument assistants for defeasible argumentation. *Artificial Intelligence Journal*, 150:291–324, 2003.
48. G. Wasson and M. Humphrey. Toward explicit policy management for virtual organizations. In *IEEE Workshop on Policies for Distributed Systems and Network (POLICY '03)*, pages 173– 182, June 2003.

How Agents Alter Their Beliefs After an Argumentation-Based Dialogue

Simon Parsons and Elizabeth Sklar

Department of Computer and Information Science, Brooklyn College,
City University of New York, 2900 Bedford Avenue, Brooklyn,
New York, NY 11210, USA
{parsons, sklar}@sci.brooklyn.cuny.edu

Abstract. In our previous work on dialogue games for agent interaction, an agent's set of beliefs (Σ) and an agent's "commitment store" (CS) — the set of locutions uttered by the agent — play a crucial role. The usual assumption made in this work is that the set of beliefs is static through the course of a dialogue, while the commitment store is dynamic. While the assumption of static beliefs is reasonable during the progress of the dialogue, it seems clear that some form of belief change is appropriate once a dialogue is complete. What form this change should take is our subject in this paper.

1 Introduction

Finding ways for agents to reach agreements in multiagent systems is an area of active research. One mechanism for achieving agreement is through the use of *argumentation*—where one agent tries to convince another agent of something during the course of some *dialogue*. Examples of argumentation-based approaches to multiagent agreement include the work of Dignum *et al.* [3], Kraus [10], Reed [15], Schroeder *et al.* [16] and Sycara [19].

The work of Walton and Krabbe [21] has been particularly influential in argumentation-based dialogue research. They developed a typology for inter-personal dialogue which identifies six primary types of dialogues including *Information-Seeking Dialogues* (where one participant seeks the answer to some question(s) from another participant, who is believed by the first to know the answer(s)); *Inquiry Dialogues* (where the participants collaborate to answer some question or questions whose answers are not known to any one participant); and *Persuasion Dialogues* (where one agent seeks to persuade another agent to adopt a belief or point-of-view she does not currently hold). This *dialogue game* [9] view of dialogues overlaps with work on conversation policies (see, for example, [2, 5]), but differs in considering the entire dialogue rather than dialogue segments.

In this paper, we extend the work of [13, 14] by considering how agents alter their beliefs as a result of participating in dialogues. In particular we are interested in the way in which the beliefs of an agent change over the course of several dialogues with another agent. The work described here allows us to obtain results which show that, under certain conditions, the beliefs of a pair of agents will converge over time.

S. Parsons et al. (Eds.): ArgMAS 2005, LNAI 4049, pp. 297–312, 2006.

2 Background

We begin by introducing the components of the formal system of argumentation that underpin our approach, as well as the corresponding terminology and notation, all taken from [1,4,13]. This is a bit lengthy, but the material is required in order to obtain the technical results later in the paper.

A dialogue game is a set of interactions that occur between two agents, M and U. Each agent maintains a knowledge base, Σ, containing formulas of a propositional language $\mathcal{L}$ and having no deductive closure. Each agent also maintains a list of utterances, called the "commitment store", *CS*. We can refer to *CS* as an agent's "public knowledge", since it contains information that is shared with other agents. In contrast, the contents of Σ are "private". The agent also maintains two Σ-like components: J and Γ. These will be discussed later. For now it suffices to know that such structures exist and are indexed by the name of the agent's dialogue partner.

Note that in the description that follows, we assume that $\vdash$ is the classical inference relation, that $\equiv$ stands for logical equivalence, and we use Δ to denote all the information available to an agent. Thus in a dialogue with U, $\Delta_M = \Sigma_M \cup \Gamma_{M,U} \cup J_{M,U} \cup CS_U$. The commitment store CS_M can be loosely thought of as a subset of Δ_M; according to the rules of the dialogue game, M can only say things it can support (or justify), i.e., using arguments in Δ_M to support propositions in CS_M.

Definition 1 (Argument). *An* argument *is a pair $A = (S, p)$ where p is a formula of $\mathcal{L}$ and S a subset of Δ such that:*

1. *S is consistent;*
2. *$S \vdash p$; and*
3. *S is minimal, so no proper subset of S satisfying both (1) and (2) exists.*

S is called the support *of A, written S = Support(A) and p is the* conclusion *of A, written p = Conclusion(A). Thus we talk of p being* supported *by the argument (S, p).*

In general, since Δ may be inconsistent, arguments in $\mathcal{A}(\Delta)$, the set of all arguments which can be made from Δ, may conflict, and we make this idea precise with the notion of *undercutting*:

Definition 2 (Undercut). *Let A_1 and A_2 be two arguments of $\mathcal{A}(\Delta)$. A_1* undercuts *A_2 iff $\exists \neg p \in Support(A_2)$ such that $p \equiv Conclusion(A_1)$.*

In other words, an argument is undercut iff there is another argument which has as its conclusion the negation of an element of the support for the first argument.

To capture the fact that some beliefs are more strongly held than others, we assume that any set of beliefs has a *preference order* over it. We consider all information available to an agent, Δ, to be stratified into non-overlapping sets $\Delta_1, \ldots, \Delta_n$ such that beliefs in Δ_i are all equally preferred and are preferred over elements in Δ_j where $i < j$. This could be thought of as saying that an agent's first choice(s) are contained in Δ_1, second choices in Δ_2, and so on. The *preference level* of a nonempty subset $S \subset \Delta$, where different elements $s \in S$ may belong to different layers Δ_i, is valued at the highest numbered layer which has a member in S and is referred to as $level(S)$.

In other words, S is only as strong as its weakest member. Note that the strength of a belief as used in this context is a separate concept from the notion of support discussed earlier. That is, a strong belief does not necessarily mean that there are many arguments supporting that belief.

Definition 3 (Preference). *Let A_1 and A_2 be two arguments in $\mathcal{A}(\Delta)$. A_1 is* preferred *to A_2 according to Pref and following the strict pre-order associated with it. In other words, $A_1 \gg^{Pref} A_2$, iff $level(Support(A_1)) \leq level(Support(A_2))$. If A_1 is preferred to A_2, we say that A_1 is* stronger *than A_2.*

We can now define the argumentation system we will use:

Definition 4 (Argumentation System). *An* argumentation system *(AS) is a triple $\langle \mathcal{A}(\Delta), Undercut, Pref \rangle$ such that:*

- *$\mathcal{A}(\Delta)$ is a set of the arguments built from Δ,*
- *Undercut is a binary relation representing the defeat relationship between arguments, $Undercut \subseteq \mathcal{A}(\Delta) \times \mathcal{A}(\Delta)$, and*
- *Pref is a (partial or complete) pre-ordering on $\mathcal{A}(\Delta) \times \mathcal{A}(\Delta)$.*

The preference order makes it possible to distinguish different types of relations between arguments:

Definition 5 (Defense). *Let A_1, A_2 be two arguments of $\mathcal{A}(\Delta)$.*

- *If A_2 undercuts A_1 then A_1* defends itself *against A_2 iff $A_1 \gg^{Pref} A_2$. Otherwise, A_1* does not defend itself.
- *A set of arguments $\mathcal{A}$* defends *A_1 iff: $\forall$ A_2 undercuts A_1 and A_1 does not defend itself against A_2 then $\exists$ $A_3 \in \mathcal{A}$ such that A_3 undercuts A_2 and A_2 does not defend itself against A_3.*

We write $\mathcal{A}_{Undercut,Pref}$ to denote the set of all non-undercut arguments and arguments defending themselves against all their undercutting arguments. The set $\underline{\mathcal{A}}(\Delta)$ of acceptable arguments of the argumentation system $\langle \mathcal{A}(\Delta), Undercut, Pref \rangle$ is [1] the least fixpoint of a function $\mathcal{F}$:

$$\mathcal{A} \subseteq \mathcal{A}(\Delta)$$
$$\mathcal{F}(\mathcal{A}) = \{(S,p) \in \mathcal{A}(\Delta) \mid (S,p) \text{ is defended by } \mathcal{A}\}$$

Definition 6 (Acceptance). *The set of* acceptable *arguments for an argumentation system $\langle \mathcal{A}(\Delta), Undercut, Pref \rangle$ is:*

$$\begin{aligned}\underline{\mathcal{A}}(\Delta) &= \bigcup \mathcal{F}_{i \geq 0}(\emptyset) \\ &= A_{Undercut,Pref} \cup \left[\bigcup \mathcal{F}_{i \geq 1}(A_{Undercut,Pref}) \right]\end{aligned}$$

An argument is acceptable *if it is a member of the acceptable set, and a proposition is* acceptable *if it is the conclusion of an acceptable argument.*

Definition 7 (Status). *If an agent M has an acceptable argument for a proposition p, then the* status *of p for that agent is* accepted*, while if the agent does not have an acceptable argument for p, the status of p for that agent is* not accepted.

An acceptable argument is one which is, in some sense, proven since all the arguments which might undermine it are themselves undermined.

3 Locutions, Attitudes and Protocols

The basis for our work is the dialogue system $\mathcal{DG}$, presented in [12] (which is a modest extension of that in [13, 14]), modified with some features from the dialogue system in [17]. Here we present as brief a summary of the combined system as we can give.

As described above, dialogues are assumed to take place between two agents, for example called M (for "me") and U ("you"). Each agent $i \in \{M, U\}$ has a knowledge base, Σ_i, containing its beliefs. We assume that this knowledge base is consistent in a certain sense — we assume that an agent only has propositions in its knowledge base for which it has an acceptable argument (the grounds of this argument may be just the proposition itself, so that, for example, an agent may have in its knowledge base p supported by the acceptable argument $(\{p\}, p)$.

In addition [9], each agent i has a further knowledge base CS_i, visible to both agents, containing *commitments* made in the dialogue. We assume an agent's *commitment store* is a subset of its knowledge base. Note that the union of the commitment stores can be viewed as the state of the dialogue at a given time. Following [17], we also assume that each agent i has a knowledge base $\Gamma_{i,j}$ where $j \in \{M, U\}, j \neq i$ which represents i's model of j's beliefs, and a set $J_{i,j}$ which records *lies* that i has told j—propositions p for which $\neg p$ is in Σ_i. Since each agent has access to their private knowledge bases and both commitment stores, agent M can potentially make use of $\langle \mathcal{A}(\Sigma_M \cup \Gamma_{M,U} \cup J_{M,U} \cup CS_U), \mathit{Undercut}, \mathit{Pref} \rangle$. For most of this paper we will assume that $\Gamma_{M,U}$ and $J_{M,U}$ are empty and so only consider Σ_M and CS_U, but towards the end we will deal with non-empty $\Gamma_{M,U}$ and $J_{M,U}$.

All the knowledge bases contain propositional formulas, and moreover all are stratified by degree of belief as discussed above. Here we assume that these degrees of belief are static and that both the players agree on them (acknowledging that this is a limitation of this approach).

During the dialogue the players put forward propositions and accept propositions put forward by other agents based on their acceptability. The exact locutions we adopt are those of [12], but for our purposes here we need only know that propositions are put forward using an *assert* locution (all the other locutions are signalling, *assert* is the only one which transmits data). The axiomatic semantics [20] of *assert* are given in Table 1. The important thing to note is that the subject of an *assert* is something that an agent either has in its knowledge base, or has an acceptable argument for, and that asserting something places it in the agent's commitment store. The *subject* of a dialogue is the argument of the first *assert* to be made—this is the proposition about which the dialogue revolves.

Table 1. Operational semantics for *assert*

assert

LOCUTION:

- $M \rightarrow U : assert(p)$

PRE-CONDITIONS:

1. $(S, p) \in \underline{\mathcal{A}}(\Sigma_M \cup CS_U)$

POST-CONDITIONS:

1. $CS_{M,i} = CS_{M,i-1} \cup \{p\}$ (update)
2. $CS_{U,i} = CS_{U,i-1}$ (no change)

The process by which a dialogue is carried out is determined by a *protocol.* An example is the protocol $\mathcal{P}''$, an extension of $\mathcal{P}'$ in [12] in which M tries to persuade U that p is the case:

1. M issues a $know(p)$, indicating it believes that p is the case.
2. M *assert*s p.
3. U *accept*s p if it has an acceptable argument for it, or U *assert*s $\neg p$ if it has an acceptable argument for that, or U *challenge*s p, or U rejects p.
4. If U asserts $\neg p$ in (3), then go to (3) with the roles of the agents reversed and $\neg p$ in place of p.
5. If U challenges in (3) then M asserts, in turn, every $s \in S$, where S is the support for p and go to (3) for each s in turn in place of p.

The "signal" locutions used here are *know*, which indicates the start of a persuasion dialogue, *challenge*, which indicates that one agent requires the other to present the support for the proposition just asserted, and *accept* and *reject*, which indicate that the agent finds (respectively, does not find) that the previously asserted proposition is supported by an acceptable argument. A signal of *accept* also indicates that the agent that issues it is no longer disputing that proposition and either the dialogue ends (if the subject of the *accept* is the subject of the dialogue), or the dialogue can pass onto the next proposition (if the subject of the *accept* is another proposition and the dialogue is the recursive phase following step 5). A signal of *reject* similarly indicates that the dialogue can pass on to the next proposition (albeit without the former propsition being accepted), and the rejection of the subject of the dialogue is the other way that a dialogue can end.

Note that, in common with previous work on this kind of system, agents are not allowed to repeat exactly the same locution in a dialogue. If the only legal move available to an agent under the protocol is to repeat itself in this way, then the dialogue terminates. This is to prevent infinite dialogues in which one agent, for example, repeatedly asserts p. By "exactly the same" we mean the same locution instantiated with a logically equivalent proposition, so that $assert(p)$ and $assert(p \wedge p)$ are considered the same locution, precisely with preventing infinite dialogues in mind (since $p \wedge p$ contains no more information that p we assume a rational agent would not *assert* both). The only exception we allow to this rule is that an agent can assert a proposition as its own grounds. Thus, as is often the case, p can be asserted as support for the previous assertion p if there is no other argument for it and p is present in the agent's knowledge base.

Note also that, for now, we don't specify how U makes the decision in step 3 of the protocol. Later we will disinguish between different ways the decision might be made and see how these relate to different outcomes.

Example 1. As an example of a dialogue that can be held under $\mathcal{P}''$, consider the following.

$\Sigma_M = \{p, p \rightarrow q\}$	M know p	
$\Sigma_U = \{p\}$	M *assert* q	$CS_M = \{q\}$
	U *challenge* q	
	M *assert* p	$CS_M = \{p, q\}$
	U *accept* p	U already has an acceptable argument for p
	M *assert* $p \rightarrow q$	
	U *challenge* $p \rightarrow q$	
	M *assert* $p \rightarrow q$	this is allowed under the exception to the repetition rule. $CS_M = \{p, q, p \rightarrow q\}$
	U accept $p \rightarrow q$	
	U accept q	

4 How Beliefs Change over Time

Previous work on argumentation-based dialogues has typically concentrated on what happens *during* a *single* dialogue — this is certainly true of the work in [12, 13, 14] — and has not contemplated what happens after a dialogue is complete, or what happens over the course of several dialogues. In contrast, our interest here is in the process by which an agent adapts its beliefs after a dialogue is ended, and what effect this process has over time. Indeed, the only related work we are aware of in an argumentation context is [11] which studies the way that beliefs change during a single argumentation-based dialogue.

4.1 Changes in Belief After a Single Dialogue

Now, without having to commit ourselves to a specific dialogue protocol, we can determine the situation that must hold at the end of a dialogue. Both of the agents engaged in the dialogue will have *assert*ed some propositions, and these will have become, in some sense, common knowledge between the two agents. Furthermore, it is clear that some of these propositions will be acceptable (in the sense of being supported by an acceptable argument) to one or both agents, and that there may be propositions p that were acceptable to an agent before a dialogue that are now no longer acceptable (because, for example, the dialogue has established that $\neg p$ is acceptable):

Proposition 1. *For any proposition p, the status of p for an agent M may change as a result of a dialogue that M has with another agent U.*

Proof. We have four cases to consider—that p is initially acceptable or not acceptable, and that p is a proposition in Σ_M or is the conclusion of an argument from Σ_M. For the result we simply have to show how the change in status may occur.

Let us assume that p is initially acceptable because it is the conclusion of an acceptable argument (S,p) where $S \subseteq \Sigma_M$ and $p \notin \Sigma_M$. The dialogue may result in U asserting an argument that undercuts the argument for p, that is an argument with conclusion $\neg s$ for some $s \in S$, and if (S,p) cannot defend itself against this argument, the status of p will change from acceptable to not acceptable.

The case for which p is initially acceptable and $p \in \Sigma_M$ is very similar. Here p is supported by the argument $(\{p\},p)$, and will change status if U asserts an argument with conclusion $\neg p$ which is preferred to $(\{p\},p)$.

If p is initially not acceptable, this is either because there is no argument that supports it, or because the supporting argument is undercut by some argument A that the supporting argument cannot defend itself against, and is not defended against by any other argument. This situation can easily change, for example if A is undercut by some newly asserted argument, and this can happen for both the case in which $p \in \Sigma_M$ and the case in which p is the conclusion of an argument (S,p) where $S \subseteq \Sigma_M$ and $p \notin \Sigma_M$.

These changes come about because the notion of acceptability is *non-monotonic*. As a dialogue between *M* and *U* proceeds, the set of propositions Δ_M that *M* uses to construct arguments increases monotonically (since no locutions remove propositions from the commitment store), but the set of acceptable arguments can both increase or decrease. (This is proved in [14][1].)

In many situations, it seems sensible for an agent to want to remember the status of the propositions that are interesting to it at the end of the dialogue. This is appropriate, for example, in our learning scenario. It might be considered less appropriate in a purchasing scenario—security might dictate that an agent should not remember sensitive data beyond the end of a dialogue. Our concern here is not on when it is appropriate to remember, but to identify mechanisms for doing so, and to explore their consequences.

There are four obvious ways to ensure that an agent *M* recalls the status of a proposition following a dialogue with *U* and these are given below. For now, we will only consider information in Σ_i and CS_i—we will come back to $\Gamma_{i,j}$ and $J_{i,j}$ later.

Definition 8 (Update Mechanisms). *We define the following* mechanisms for updating *Σ_M at the end of a dialogue between agents M and U.*

W1: Expand Σ_M to become $\Sigma_M \cup CS_U$.

W2: Expand Σ_M with all s for which there exists a p such that $(S,p) \in \underline{\mathcal{A}}(\Sigma_M \cup CS_U)$, $s \in S$ and $s \notin \Sigma_M$.

[1] And can be easily seen in the following example. *M* initially has just one argument $(\{q, q \rightarrow p\}, p)$ for *p*, and by definition this is acceptable. *U* then puts forward the argument $(\{r, r \rightarrow \neg p\}, \neg p)$ for $\neg p$. Both agents only have knowledge bases that consist of the support of their arguments, and all propositions are equally preferred. After the second argument is asserted, neither argument is acceptable to either agent, and so *M*'s set of acceptable arguments has shrunk while its set of arguments has grown.

W3: Expand Σ_M with all logically distinct p such that $(S, p) \in \underline{\mathcal{A}}(\Sigma_M \cup CS_U)$ and $S \not\subseteq \Sigma_M$.

W4: Replace any $p \in \Sigma_M$ such that $(S, \neg p) \in \underline{\mathcal{A}}(\Sigma_M \cup CS_U)$ with $\neg p$.

Of course, though we have stated the update mechanisms for M alone, there are symmetrical mechanisms for U.

In other words, these mechanisms are as follows: (1) add everything in U's commitment store to M's knowledge base[2]; or (2) add those elements of the support of propositions p for which M only has an acceptable argument *after* the dialogue; or (3) add just the propositions p for which M only has an acceptable argument *after* the dialogue; or (4) replace any propositions in Σ_M whose negations are now acceptable with those negations.

In conjunction with Definition 8, we need to define what constitutes a good mechanism for this updating. It seems reasonable to insist that the update is to ensure that the agent in question keeps a record of just those new propositions that it finds acceptable.

Definition 9 (Update Criteria). *We define the following* criteria for updating *the knowledge-based Σ_M of agent M after a dialogue:*

C1. Updating should cause the addition to Σ_M of exactly those propositions that are acceptable at the end of the dialogue but were not acceptable before the dialogue began.

C2. After updating, $\underline{\mathcal{A}}(\Sigma_M)$ should include all those arguments that are acceptable at the end of the dialogue.

We can use these criteria to identify which mechanism for updating should be adopted, but first we need:

Lemma 1 (from [13]). *If $(S, p) \in \underline{\mathcal{A}}(\Sigma_M)$ then $(S', s) \in \underline{\mathcal{A}}(\Sigma_M)$ for every $s \in S$.*

In other words, every element of the support of an acceptable argument is itself the conclusion of an acceptable argument.

Corollary 1. *If an updating mechanism satisfies C1, then it satisfies C2.*

Proof. Immediate from the definition of C1 and C2, and Lemma 1.

Thus C1 is a stronger criterion than C2 since it specifies that no additional propositions other than those that have become newly acceptable should be added. C2 allows for the addition of propositions that result in Σ_M generating arguments after the updating that are not acceptable so long as all arguments that were acceptable at the end of the dialogue can be constructed. Thus C2 does not imply C1[3].

Proposition 2. *Mechanisms W1 and W2 satisfy C2, mechanism W3 satisfies C1, and mechanism W4 fails to satisfy either criterion.*

[2] This is just the simplest update rule we can imagine, rather than one we think would be adopted by a rational agent, but would be a possible update rule for the *credulous* agents discussed in [13].

[3] We see no way of tightening C2 to make $\underline{\mathcal{A}}(\Sigma_M)$ generate exactly the arguments acceptable at the end of the dialogue without losing valuable information.

Proof. *We examine each mechanism in turn, considering the case of updating Σ_M after agent M has completed a dialogue with agent U.*

W1 updates by adding every proposition in CS_U to Σ_M. If M has asserted some proposition that M does not find acceptable, then this will be added to Σ_M (since all propositions asserted by U end up in CS_U whether or not M finds them acceptable). W1 thus fails to meet C1 by including propositions that M does not find acceptable, but by adding everything that was asserted by U satisfies C2—all new arguments, including all the acceptable ones, can be constructed.

W2 updates by including the grounds for every p that has become acceptable as a result of the dialogue, and so satisfies C2. It fails to satisfy C1, however, because it does not include the p themselves (unless they are in the grounds of other acceptable arguments).

W3 updates by adding to Σ_M every logically distinct conclusion of every acceptable argument whose support is not already wholly in Σ_M. Since Lemma 1 tells us that every element of the support of such arguments will also be the conclusion of an acceptable argument, the result will be to include all formulae that are acceptable after the dialogue but were not before, which exactly satisfies C1.

W4 updates by replacing every p in Σ_M that was acceptable before the dialogue but is not afterwards by $\neg p$. This is in line with C1 for those propositions which were acceptable before the dialogue and have become unacceptable as a result of it, but fails to deal with propositions for which there was no argument before the dialogue. W4 thus satisfies neither C1 nor C2.

Given this result, the most suitable of these procedures for revision seems to be W3, since it satifies the strongest of the conditions, though examining the proof of Proposition 2 shows that W2 is very nearly as good.

As an illustration of how W3 works, consider the following.

Example 2. After the dialogue in Example 1, U will add $p \rightarrow q$ and q to Σ_U since there are acceptable arguments for these, and the grounds for the argument were not all previously in Σ_U. M will add nothing to Σ_M since U asserted no propositions, and so there are no new arguments that are acceptble to M — note that M does not add q even though it is not part of its original knowledge base.

4.2 Changes in Belief over Several Dialogues

Our primary interest in this paper is to examine how the knowledge-bases of agents develop over time, which we measure in terms of a series of dialogues. To track this development we need the following definition:

Definition 10 (Degree of Agreement). *The* degree of agreement *DA between two sets of formulae S_1 and S_2 is:*

$$DA(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

Thus we define the agreement between two knowledge bases by looking at the proportion of formulae they have in common. Two knowledge bases which share no formulae will have a *DA* of 0, and two knowledge bases which contain exactly the same set of formulae will have a *DA* of 1. Note that the measure as defined is symmetrical and makes no attempt to identify whether one knowledge base is contained in another, a situation that could be considered another form of agreement. We acknowledge that more sophisticated measures of agreement can be established, but this seems to suit our requirements for now.

Given Definition 10 we can establish how a given dialogue changes the extent to which two agents agree. It is simple to show that:

Proposition 3. *If M and U engage in repeated $\mathcal{P}''$ dialogues and update using W3 after each, then the degree of agreement between Σ_M and Σ_U may not increase.*

Proof. For this proof it suffices to show that there is a way for the degree of agreement to not increase.Consider that M starts a dialogue by asserting p, U challenges, and M asserts the support for its argument (S,p). If U rejects the first $s \in S$, then at the end of the dialogue neither agent has anything to add to its knowledge base. This same process can happen for every dialogue, and the degree of agreement between Σ_M and Σ_U will not increase.

There are several comments to make about this result. The first is that the result captures an extreme case—over many dialogues it seems likely that at least one proposition will be accepted by one agent, and so the degree of agreement will increase a little. However, the point the proposition makes is that there is no guarantee that it will. The second comment is that this can be viewed as a good thing. Gabbay and Woods in their discussion of non-coooperation dialogues [6, 7] give the example of a police interrogation, where it may very much be in an agent's interests not to be persuaded that something is true (that one committed a crime about which one has no knowledge for example).

The main comment to make about this result is that though it is weak — it just says that after some dialogues the agents might not be any closer to agreement — the reason behind it suggests the subject deserves more investigation. The reason that agents might not have a greater degree of agreement after a dialogue is, as it is easy to see from the proof of Proposition 3, that if *U* finds *s*, that is some part of *M*'s support for (S,p), unacceptable, it can just *reject* and end the dialogue. This can happen even if *M* has information that would overturn *U*'s objection to *s* if it were stated. It is this latter possibility that seems worthy of elucidation, especially when we realise that the property of resisting an increase in agreement is not just a property of $\mathcal{P}''$, but also of the various kinds of dialogue introduced in [13].

If we define:

Definition 11 (Closed Mouth Dialogue). *A dialogue between two agents M and U is a* closed mouth dialogue *if either agent replies to an "assert(p)" with an immediate "accept(p)" or "reject(p)" during the course of the dialogue.*

Definition 12 (Open Mouth Dialogue). *A dialogue between two agents M and U is an* open mouthed dialogue *if both agents can only reply to an "assert(p)" with "challenge(p)" or "assert$(\neg p)$" before "reject(p)".*

As introduced the protocol $\mathcal{P}''$ can generate both open-mouthed and a closed-mouth dialogues, but we can devise open and closed mouth versions of $\mathcal{P}''$ that can, respectively, only generate open and closed mouth dialogues. One closed-mouth variant of

$\mathcal{P}''$, denoted $\mathcal{P}''_{CM}$, rejects *whenever* the asserted proposition is not acceptable[4] accepting otherwise. The open-mouthed variant, denoted $\mathcal{P}''_{OM}$, challenges whenever the asserted proposition is not acceptable unless such a challenge would be a repetition. When it cannot challenge, the agent asserts the negation of the asserted proposition if that is possible, and can only accept or reject when such an assertion is impossible. Finally $\mathcal{P}''_{OM}$ accepts if the proposition is acceptable and rejects otherwise.

We are now nearly at a point where we can relate the form of the dialogue, open or closed-mouth, to degree of agreement. Before we can make such a relation, however, we need to consider that each agent "updates" its knowledge base Σ_i with the conclusions p of all acceptable arguments (S, p) that can be made from Σ_i (in other words the agents add every p such that $(S, p) \in \underline{\mathcal{A}}(\Sigma_i)$), doing a kind of pre-emptive W3 update.

With this condition, then, we have:

Corollary 2. *If M and U engage in any series of dialogues under $\mathcal{P}''_{CM}$ and update using W3 after each, then the degree of agreement between Σ_M and Σ_U will not increase.*

Proof. The proof follows quickly from 3. If U rejects whenever the proposition is unacceptable, the only time it can possibly accept is if the proposition is immediately acceptable, but in that case U must have an acceptable argument for it before the dialogue starts, and so the degree of agreement will not increase.

which makes the point that some closed mouth dialogues (the example we have the result for is only one example of a closed mouth dialogue) prevent two agents increasing their degree of agreement. If we didn't add the condition on the knowledge bases before the dialogue, of course, then the degree of agreement would increase if *M*'s assertion made *U* "realise" that it had grounds to support *p* all along but just hadn't generated an argument for *p*.

The key thing about an open-mouthed dialogue is that each agent has to explain why it finds a proposition *p* unacceptable, challenging if it doesn't have enough information to construct a support for it, and asserting $\neg p$ if it has an argument against it. This results in:

Proposition 4. *At the end of a dialogue about p under $\mathcal{P}''_{OM}$ between agents M and U, p must have the same status for M and U.*

Proof. By definition, in an open mouthed dialogue, if one agent does not have an acceptable argument for a proposition p asserted by the other, it has to either challenge, which will lead to the assertion of other propositions, or assert $\neg p$, which will result in a challenge and the assertion of the grounds for $\neg p$. This process will recurse until neither agent has anything more challenges or assertions to make, and all the information which either agent can bring to bear on the subject has been deployed. At this point both agents have access to the same set of arguments concerning every p asserted by both agents (otherwise the recursion would not have stopped), and both agents will have to grant every p that has been asserted the same status.

This result takes us close to being able to identify open-mouthed dialogues with increases in the degree of agreement, but we first have to consider cases like that in the following example:

[4] Other closed mouth variants of $\mathcal{P}''$may immediately reject some assertions and not others.

Example 3. All propositions have the same preference level:

$$\begin{array}{lll}
\Sigma_M = \{p \wedge q\} & M \text{ know } p & \\
\Sigma_U = \{p \wedge \neg q\} & M \text{ assert } q & CS_M = \{q\} \\
 & U \text{ challenge } q & \\
 & M \text{ assert } p \wedge q & \\
 & U \text{ challenge } p \wedge q & \\
 & M \text{ assert } p \wedge q & \\
 & U \text{ assert } \neg q & CS_U = \{\neg q\} \\
 & M \text{ challenge } \neg q & \\
 & U \text{ assert } p \wedge \neg q & \\
 & M \text{ reject } \neg q &
\end{array}$$

Here agreement on the status of q means both find q unacceptable — M has an argument for q, but it is undercut by the $\neg q$ in CS_U, U has an argument for $\neg q$ but this is undercut by the q in CS_M — and so neither will update its Σ. This is the kind of situation in which, in human argumentation, we say “we must agree to disagree”. Both sides have heavily entrenched beliefs that lead to inconsistent positions that cannot be resolved. We capture this in the notion of *deadlock*:

Definition 13 (Deadlock). *Two agents M and U are* deadlocked *over p if $(S,p) \in \underline{\mathcal{A}}(\Sigma_M)$ or $(S,p) \in \underline{\mathcal{A}}(\Sigma_U)$, but $(S,p) \notin \underline{\mathcal{A}}(\Sigma_M \cup \Sigma_U)$.*

The notion of deadlock captures exactly the case in the example above as well as the case where both M and U initially have an acceptable argument for p, but these arguments are built on contradictory grounds — the grounds will be exposed by the dialogue, and neither agent ends up finding the subject of the dialogue acceptable (of course, in such a case one might want to revise not just by W3, but by removing some propositions, but we will leave such considerations for future work — the system we deal with here would just end the dialogue with the contradiction unresolved in such a case).

Proposition 4 captures the limits of persuasive argumentation, at least as far as open-mouthed dialogues are concerned. In an open-mouth dialogue each agent says all that it has to say relating to a subject, but that does not guarantee to create agreement. However, we have a more “agreeable” result if agents are not deadlocked:

Proposition 5. *If two agents M and U engage in a dialogue under $\mathcal{P}''_{OM}$ with subject p, and update using W3 after, then the only cases in which the $DA(\Sigma_M, \Sigma_U)$ does not increase as a result of the dialogue is when either (1) the agents both initially have the same acceptable argument for p or (2) the agents are deadlocked over p and all the grounds for p that M asserts.*

Proof. Consider the progress of an open mouth dialogue as sketched in the proof of Proposition 4. There are only two ways that this process will not lead to some new propositions being accepted by one of the agents, thus increasing $DA(\Sigma_M, \Sigma_U)$. One way is if every assertion is met with an accept. For this to be the case, the two agents must have exactly the same argument for p (and it must be acceptable or otherwise it could not be asserted by either). The other way is if every assertion is ultimately met *with a reject, and that can only happen if the agents are deadlocked on every proposition that is asserted — p and every proposition that is in the grounds for p that are asserted by M.*

Thus over many dialogues, we can say that the knowledge bases of the two agents will converge—if they talk for long enough, then they will agree:

Proposition 6. *If M and U engage in n successive dialogues under $\mathcal{P}''_{OM}$ with different subjects, update using W3 after, and are not deadlocked about any of the assertions made during the dialogues, then:*

$$\lim_{n \to \infty} DA(\Sigma_M, \Sigma_U) = 1$$

Proof. Under the conditions stated, Proposition 5 tells us that for each dialogue, either the degree of agreement will increase after that dialogue, or the agents already had the same acceptable argument for the subject of the dialogue. Since the subject changes after each dialogue, this means that as $n \to \infty$, either the degree of agreement increases monotonically, or the agents had exactly teh same set of propositions to begin with (and so had the same acceptable argument for every *subject). In the former case the degree of agreement increases to 1, in the latter case it was 1 to begin with.*

We need the condition about the dialogues having different subjects to prevent the case in which the agents keep having the same dialogue (or small finite set of dialogues) and the degree of agreement never moves beyond some value $\epsilon < 1$. In addition, as the proof points out, there is a degenerate case of "convergence" in which the two agents started out with identical knowledge bases. However, except for this case the convergence is real, and seems likely to be quick. Given Proposition 4, we know that the degree of agreement of the agents will increase by at least one proposition (the subject of the dialogue) each time, and so convergence will require at most N rounds of dialogue, where $N = |\Sigma_M \cup \Sigma_U|$[5]. Finally, we should mention that the condition on deadlock is required for the theorem as stated, but might be relaxed without serious effect on what happens in real dialogues — if the agents are deadlocked on some set of propositions, but this set is small compared with $|\Sigma_M \cup \Sigma_U|$, then the degree of agreement will approach 1.

4.3 Lying and Modelling Other Agents

The results so far concentrate on changes to Σ_M and Σ_U. We can also derive convergence results for the sets of lies each agent has told, $J_{M,U}$ and $J_{U,M}$, and for the models each agent has of the other, $\Gamma_{M,U}$ and $\Gamma_{U,M}$. Let's start by considering $\Gamma_{M,U}$ and $\Gamma_{U,M}$, and extend our update procedure W3 so that at the end of a dialogue with U, M not only updates its knowledge base Σ_M with all the propositions p for which it has an acceptable argument, but also updates its explicit model of U with information it knows that U now accepts. With this additional information $\Delta_M = \Sigma_M \cup \Gamma_{M,U} \cup CS_U$.

[5] Note though that convergence will require both agents to carry out some persuasion — recall that in Example 2, M did not add q to its knowledge base. For q to be accepted by M, U would have to assert q in some later dialogue.

We need some additional definitions:

Definition 14 (Sound Model). *If $\Gamma_{M,U}$ is the model M has of the beliefs of U, then it is a* sound model *of U if $p \in \Gamma_{M,U}$ iff $p \in \Sigma_U$.*

Definition 15 (Complete Model). *If $\Gamma_{M,U}$ is the model M has of the beliefs of U, then it is a* complete model *of U if $p \in \Sigma_U$ iff $p \in \Gamma_{M,U}$.*

With these we can extend Proposition 6 to get:

Proposition 7. *If M and U engage in n successive dialogues under $\mathcal{P}''_{OM}$ with different subjects, and update using W3 after, then as $n \rightarrow \infty$, $\Gamma_{M,U}$ becomes a sound and complete model of U.*

Proof. Clearly $\Gamma_{M,U}$ is sound and complete if $DA(\Gamma_{M,U}, \Sigma_U) = 1$. Since updating $\Gamma_{M,U}$ takes place in the same way as updating Σ_M, the result follows directly from Proposition 6.

Thus if they talk for long enough, one agent will converge on a sound and complete model of the other's beliefs. As our discussion of Proposition 6 argues, the number of dialogues required for this convergence is linear in the size of the agents' knowledge bases.

Finally, for agents that are lying, we need to add in the $J_{i,j}$ so that $\Delta_M = \Sigma_M \cup J_{M,U} \cup CS_U$. Recall that the idea of $J_{M,U}$ is that it records things that *M* believes are false, but uses to build arguments that it seeks to persuade *U* with — the arguments are not acceptable to *M* (and in [18] we introduce new semantics for *assert* to deal with this) — and records in order to attempt to only assert things to *U* that are consistent with the contents of $J_{M,U}$. In such a situation, what *M* wishes to avoid is being caught in a lie:

Definition 16 (Caught in a Lie). *An agent is* caught in a lie over *p if it is forced to assert both p and $\neg p$ in the same dialogue.*

We have to define being caught in a lie like this, rather than, for example, as the assertion of *p* and $\neg p$ in different dialogues, since an agent may do this innocently, having changed the status of *p* in between.

Proposition 8. *If M and U engage in a n successive dialogues under $\mathcal{P}''_{OM}$ with different subjects, then if M lies to U about p and the probability of M being caught in a lie over p is denoted by $\Pr(c(p))$, then:*

$$\lim_{n \rightarrow \infty} \Pr(c(p)) = 1$$

Proof. If M is in an open-mouthed dialogue with U, M always has to back up its position on every proposition p, and this involves stating the support S, where S may be drawn from Σ_M or $J_{M,U}$. Given what U utters, there is some probability that a given proposition k will be required to be asserted as such support, P_k (we allow this to vary from proposition to proposition). Assuming the probabilities of needing to assert p and $\neg p$ are independent, the probability that M will be caught in a lie is thus $\Pr(c(p)) = P_p \cdot P_{\neg p}$ (which may be very small), and so the probability of not being caught is $1 - P_p \cdot P_{\neg p}$, which is, by definition, less than 1. After n dialogues, the probability of not being caught, $1 - \Pr(c(p)) = (1 - P_p \cdot P_{\neg p})^n$, and this will converge to 0 as n tends to ∞. Thus the result holds.

Indeed, the result holds even if P_p and $P_{\neg p}$ are not independent—simply replace $P_p \cdot P_{\neg p}$ with $P_{p,\neg p}$, and so long as this is not zero, as long as it is possible that M will be caught, the probability of being caught converges to 1 as the number of dialogues increases.

In other words, the more dialogues that M and U engage in, the greater the chance that M will be caught in a lie. This result depends only on the properties of $\mathcal{P}''_{OM}$ (in a dialogue under $\mathcal{P}''_{CM}$, M would not have to produce grounds) and not the properties of any update operator.

5 Conclusions

This paper has extended the work of [14], which identified the range of possible outcomes of argumentation-based dialogues. Here we have considered what happens at the end of a dialogue—that is what mechanisms are suitable for altering an agent's record of what it believes as a result of a dialogue—and how what happens at the end of a dialogue impacts how an agent's beliefs change after a sequence of dialogues. Our main result is that the way the beliefs change over this sequence depend on the properties of the dialogues themselve, and under certain circumstances, the beliefs of two agents tend to converge as the number of dialogues they engage in grows.

There are three ways that we intend to pursue extensions to this work. One is to consider the mechanisms we have for updating beliefs at the end of a dialogue from the perspective of belief revision [8]. The mechanism we proposed here can clearly be considered as a belief revision mechanism, the question is whether it conforms to the standard properties for such a mechanism. The second extension we plan is to work back towards the results obtained in [14]. That work, in contrast to ours, considered the results of just a single dialogue, and made precise predictions about the outcome based on the contents of the participating agents' knowledge bases. Our work looks at the outcomes of a sequence of dialogues in very general terms, and we would like to see if we can make more precise predictions if we look at the contents of the particpants' knowledge bases in more detail. Finally we intend to look at other forms of open and closed mouth dialogues — the ones we have considered here are two variants of a single protocol — seeking to identify what properties hold for open and closed mouth dialogues in general.

Acknowledgments. This work was partially supported by NSF REC-02-19347, NSF IIS-0329037, and EU PF6-IST 002307 (ASPIC), and has benefited from conversations with Leila Amgoud, Eva Cogan, Peter McBurney and Mike Wooldridge. We are grateful to anonymous reviewers of a previous version of this paper for helping us to clarify our thoughts on this subject.

References

1. L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation framework. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 1–7, 1998.

2. B. Chaib-Draa and F. Dignum. Trends in agent communication language. *Computational Intelligence*, 18(2):89–101, 2002.
3. F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In C. Castelfranchi and Y. Lespérance, editors, *Seventh Workshop on Agent Theories, Architectures, and Languages*, pages 141–156, Boston, USA, 2000.
4. 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.
5. R. A. Flores and R. C. Kremer. To commit or not to commit. *Computational Intelligence*, 18(2):120–173, 2002.
6. D. M. Gabbay and J. Woods. More on non-cooperation in Dialogue Logic. *Logic Journal of the IGPL*, 9(2):321–339, 2001.
7. D. M. Gabbay and J. Woods. Non-cooperation in Dialogue Logic. *Synthese*, 127(1-2): 161–186, 2001.
8. P. Gärdenfors. *Knowledge in Flux*. MIT Press, 1988.
9. C. L. Hamblin. Mathematical models of dialogue. *Theoria*, 37:130–155, 1971.
10. S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.
11. P. McBurney and S. Parsons. Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and Artificial Intelligence*, 32(1–4):125–169, 2001.
12. S. Parsons, P. McBurney, and M. Wooldridge. Some preliminary steps towards a meta-theory for formal inter-agent dialogues. In Iyad Rahwan, editor, *Proceedings of the 1st International Workshop on Argumentation in Multiagent Systems*, New York, 2004.
13. S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *1st International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 2002.
14. S. Parsons, M. Wooldridge, and L. Amgoud. On the outcomes of formal inter-agent dialogues. In *2nd International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 2003.
15. C. Reed. Dialogue frames in agent communications. In Y. Demazeau, editor, *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 246–253. IEEE Press, 1998.
16. M. Schroeder, D. A. Plewe, and A. Raab. Ultima ratio: should Hamlet kill Claudius? In *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 467–468, 1998.
17. E. Sklar and S. Parsons. Towards the application of argumentation-based dialogues for education. In N. R. Jennings, C. Sierra, E. Sonenberg, and M. Tambe, editors, *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems*. IEEE Press, 2004.
18. E. Sklar, S. Parsons, and M. Davies. When is it okay to lie? a simple model of contraditcion in agent-based dialogues. In *Proceedings of the First Workshop on Argumentation in Muliagent Systems*, 2004.
19. K. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the Eleventh Joint Conference on Artificial Intelligence*, pages 517–523, 1989.
20. R. D. Tennent. *Semantics of Programming Languages*. International Series in Computer Science. Prentice Hall, Hemel Hempstead, UK, 1991.
21. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, USA, 1995.

Author Index

Aguirre, José L. 279
Amgoud, Leila 88, 217, 264

Belabbes, Sihem 264
Boella, Guido 29
Brena, Ramón F. 279

Chesñevar, Carlos I. 279
Cogan, Eva 154

Dimopoulos, Yannis 169

Eemeren, Frans H. van 1

Fischer, Felix 122

Houtlosser, Peter 1
Hulstijn, Joris 29
Hunter, Anthony 245

Jennings, Nicholas R. 104

Kakas, Antonis C. 169
Karunatillake, Nishan C. 104

McBurney, Peter 42, 154
McGinnis, Jarred 199

Modgil, Sanjay 57
Moraitis, Pavlos 169

Norman, Timothy J. 104

Parsons, Simon 42, 154, 229, 297
Prade, Henri 88, 264
Prakken, Henry 138

Rahwan, Iyad 104, 122
Reed, Chris 74
Robertson, David 199
Rovatsos, Michael 122

Sklar, Elizabeth 297

Tang, Yuqing 229
Torre, Leendert van der 29

Veenen, Jelle van 138
Vreeswijk, Gerard A.W. 182

Walton, Chris 199
Weiss, Gerhard 122
Wells, Simon 74
Wooldridge, Michael 42