# Towards An Arguing Agents Competition: Architectural Considerations

**Simon Wells**[1] and **Paweł Łoziński**[2] and **Minh Nhat Pham**[3]

**Abstract.** The primary purpose of this paper is to contribute to the discussion regarding the nascent Arguing Agents Competition (AAC) by exploring some of the issues relevant to the creation of a distributed online Competition. A secondary goal is to outline the implementational work that is currently underway and both to spur interest in contributions towards the development of the AAC architecture and to elicit participation in the eventual, resulting competitions. To achieve these ends an architecture for supporting a persistent, distributed, online AAC is introduced that is rooted in an extant computational argumentation game. Following this some issues relevant to such an approach are explored.

## 1 INTRODUCTION

The Arguing Agent Competition (AAC) is seen as an open environment in which heterogeneous agents can engage in competitive argument against one another. The notion is that agents can be constructed that argue with each other, according to a specific argumentation game protocol, and the assumption is made that the agent with the best strategy will "win" according to the criteria of the competition, therefore giving us a way to identify successful argumentation strategies. Given this notion of an AAC, the main aim of this paper is to explore the architecture necessary to support a distributed, online AAC in the context of work already carried out, and currently being extended at a number of locations. The initial starting point is to adopt *Argumento*, an extant argumentation game engine, as the argument-game playing core, and to extend this with new functionality to support a persistent online argument competition. The current state of *Argumento* and its place in the proposed AAC is explored more fully in a companion paper [28] which explores the motivation for an AAC, the motivation for the adoption of *Argumento* as an intial core game playing engine, and proposed enhancements that are required to be made to *Argumento* to support an AAC. It should be noted that *Argumento* is not a comprehensive computational argumentation game engine and there are many other approaches to argumentation which could be explored in the context of an AAC, but by using an existing game implementation initially, more rapid progress can be made on developing the distributed architecture because there is one less aspect to implement initially. Rather, the notion o supporting a variety of game engines using a pluggable architecture is envisaged which would enable a degree of flexibility in the implemented AAC. However by adopting *Argumento*, the basic system can be used as a scaffold to support the more rapid development of the distributed aspects of the AAC. Once a distributed architecture is running, then *Argumento* can be enhanced, or even replaced, to support different models of argumentative interaction enabling different approaches to argumentative interaction to be explored. By having a live online system, in which agents are continuously engaging in competitive argumentative behavior, and into which new agents, carrying new argumentative strategies, can be injected at any time, a continuous agent argument benchmarking system can be built. This could be used to measure some aspects of progress in the field of computational argument, but also to spur, or at least facilitate friendly competition between researchers in the field. This paper examines a possible architecture for a persistent, distributed AAC and explores some aspects of both monologic and dialogic argument as they pertain to such a framework.

[1] University of Dundee, U.K., email: swells@computing.dundee.ac.uk
[2] Warsaw University of Technology, Poland, email: pawel.lpl@gmail.com
[3] University of Dundee, U.K., email: m.n.pham@dundee.ac.uk

## 2 BACKGROUND

This papers builds on three ideas, firstly, that an existing argument application could be expanded to form the foundation of an AAC. Secondly that there is an existing model for competitive agent development that could inform the design of an AAC. Finally, that there are currently a variety of approaches to doing computational argumentation and that an AAC should support at least the most common approaches if not a wider variety of approaches.

### 2.1 *Argumento*

*Argumento* is an extant software system [29] for playing computational argument games that enables agents to play an *argumentation game* using an abstract argumentation system modeled after Dung [9] and an argument game adapted from Wooldridge [27]. *Argumento* currently enables human-human, human-agent, and agent-agent play and it is the automated agent-agent play that is of interest when building an AAC. In this paper's companion [28] a number of extensions are discussed that could be used to help *Argumento* to form the basis of an AAC. These include enhancements to the existing argument game, enforcement of move legality, adding and removing agents from the competition, simplifying the running of group competitions, enabling game review, and inclusion of multiple dialogue games.

However *Argumento*, even if all of the proposed extensions are implemented, is still a centralized, standalone application that is not networked, and is oriented towards a one-off competition situation. A more enticing vision is that of a persistent online AAC where the online repository continues the centralized refereeing activities that are currently within *Argumento*'s remit, but where the arguing agents themselves can be distributed about many different systems. Ideally there would be a constant source of new agents with live statistics available through a *spectator* web-interface indicating which agents are performing best in the competition at any given point.

## 2.2 TRADING AGENT COMPETITION

One place to start in planning the construction of an AAC is to look at existing competition systems. The most popular is arguably the Trading Agent Competition (TAC) [26] held at the annual Autonomous Agents and Multiagent Systems (AAMAS) conference. The aim of the TAC is to promote and encourage research into trading agents by making available an arena in which researchers can pit their creations against each other in a competitive fashion.

TAC is currently centered around two competitions, TAC classic [25], based upon the travel agent scenario in which agents compete in multiple simulataneous auctions to complete a complex procurement process, and TAC SCM [3], a supply chain management competitions using a computer manufacturer scenario in which agents compete to source components, and build and sell machines. Clearly, the success of TAC, both in terms of the number of competitors regularly taking part and the number of papers published partly as an outcome from developing TAC agents, is something that the AAC should aim to emulate. There is the question though of how the TAC model can be improved upon. A TAC is run each year at which the TAC champion is identified and developers can download and run their own TAC servers, however the provision of public TAC servers is lackluster with no facility to access records of competitions or to extract statistical data.

Ideally therefore, the AAC would be played in a number of modes. Primarily, there would be the flagship mode in which an annual competition is played wherein competing agents argue, and the best is crowned the champion. In the day-to-day mode, agents could either be uploaded to an AAC server on which they are run, possibly in a virtual machine for safety, or could be run in a distributed fashion, joining suitable competitions run by the AAC server as they are announced. Finally it should be a straightforward matter to download and setup a new server to allow *ad hoc* offline competitions to be run. Additionally a web interface to previous competitions allowing retrieval of statistical data about the AAC might prove to be a useful source of research data.

## 2.3 ARGUMENTATION IN AGENTS & MAS

Investigations into intelligent, autonomous agents, and multiagent systems (MAS) have generally looked at two facets of argumentation, the first investigates aspects of defeasible and automated reasoning and looks at how agents can use arguments either internally as a part of their decision making processes, or externally, to determine how to evaluate another agents position, and the second facet investigates how argumentation based interaction protocols can be exploited in inter-agent communication.

Argumentation frameworks are one way in which argumentation has been adopted for use in agents. Currently the most popular framework in MAS seems to be the Dung Argumentation Framework (DAF) [9], which identifies a network of attack relationships between arguments in a graph. From this graph various attributes for any given argument, or set of arguments, can be computed which allow conclusions to be drawn about the graph and the arguments that populate it. There has been much work recently to adapt and extend DAFs to incorporate necessary argumentation theoretic concepts such as support relationships into the framework.

A popular approach to specifying agent interaction is in terms of a simple *Dialectical Game*, a term coined by Hamblin [15]. Dialectical games are, in the simplest case, basically two-player, turn-taking games which are used to structure the interactions between a dialogue's participants who use their turn to make moves which correspond to the kinds of things that they can say for example, responding, affirming, questioning, &c. It is the rules of the dialectical game that specify which moves are legal or illegal at any given point in a dialogue and also how making any given move affects the positions of the participants as a result, for example, by altering the speakers *dialogical commitment* with respect to whatever was said.

Dialectical games have been proliferating recently and have been utilized in a wide range of applications including, to refer to but a few, the analysis of logical fallacies [15], as tools for belief revision in AI systems [11], in computer based learning to structure student-tutor interaction [18], in law to explore evidential structures [5], and in multiagent systems to structure communicative agent interaction [17].

This raises the question of the kinds of argumentation that an AAC should support. Currently *Argumento* supports a Dungian DAF on which a game is played but, given the variety of frameworks for representing arguments, it would be prudent to enable flexibility in the choice of framework used during a competition. The benefit of enabling such a flexibility might also be felt in the adaptability and utility of any argumentation strategies that are subsequently developed if they are designed to perform well under any argumentation framework rather than just exploiting the features of a single given framework. Interaction protocol representation in *Argumento* is currently rudimentary and is predicated on features of the underpinning abstract argumentation framework, such as movement from one argument node to another in the argument graph along the attack relationship edges. Many dialectical games, although themselves designed to be simple, are much more sophisticated than the *Argumento* game in terms of the range of moves and generally support a range of moves, each of which is usually fashioned after some form of illocutionary utterance[4] or speech act [20]. These moves generally enable a participant to explore a more complex network of arguments than the simple attack networks of Dung and should therefore be necessary to enable interesting games to be played upon more complex argumentation frameworks.

# 3 DISTRIBUTED ARCHITECTURE

The aim is to construct a distributed online software system that supports an AAC whilst also talking advantage of existing software so as to minimize new development in the first instance. By adopting the extant *Argumento* software, the AAC has a scaffolding that provides early functionality but with the disadvantage of only being able to play local games, e.g. on the same machine that the *Argumento* software is running on. Currently new *Argumento* agents are created by extending the *Argumento Agent.java* class and implementing method to account for various agent behaviours. A simple way to distribute this would be to enable *Argumento* agents to run on remote machines, however this would have the disadvantage of still requiring agents to be written using the *Argumento* agent class, when there are many popular agent development framework available that are much more advanced. The proposed solution is to create a proxy agent that appears to *Argumento* to be a normal *Argumento* agent but which forwards all of the communcations between the *Argumento* engine and the system over the internet via a web service based infrastructure to a remote agent system. The current plan is to create an application programming interface to support the development of *Argumento* agents in a variety of agent platforms, including JUDE [16], JACK [14], and JADE [10]. The webservice architecture is planned also to be written in Java and supported using Apache Tomcat [2] to

support the web applications and Apache Axis 2 [1] to support the web services infrastructure. This architecture is illustrated in figure 1 which shows the AAC server containing an *Argumento* engine and two native *Argumento* agents. A number of *Argumento* agent proxies are also illustrated connecting the *Argumento* engine, which sees each proxy merely as another *Argumento* agent, though a web service interface to other agents, one for each proxy, developed using alternative agent frameworks. In the illustration agents three through eleven are non-native, proxied *Argumento* agents. The advantage of this approach is that each agent is able to take advantage of the resources in their own implementation framework, for example, the JUDE agents six, seven, and eight, could each make use of the JUDE defeasible reasoner to support their individual, internaly reasoning and deliberation processes, before communications are serialised over the web service interface back to the *Argumento* engine for appraisal according to the game rules.

## 4 MONOLOGIC ARGUMENTATION ISSUES

In this section we will explore the effect that extending [29] Argumento to distributed architecture has on the *Arguing Agents Competition*. To state the obvious, all the problems generated by a distributed approach to argumentation can be to a great extent simulated in a centralized system running on a single machine, but in a distributed environment one can no longer choose whether to invoke them and deal with them or not.

AAC, as any distributed system, faces the standard issues that occur in such architecture as a result of its characteristic features: resource sharing, openness, concurrency, scalability, fault tolerance and transparency. It is not the purpose of this paper to review these issues as it is the field of separate study. Rather, we focus on some of the main issues that distribution cause from the dialogical perspective.

### 4.1 Basic issues

The very basic problem that needs to be addressed is explicit formulation of the assumptions regarding the knowledge that agents have about the argumentation graph that underlies their discussion. In [29] the assumption is that agents know about the whole graph, which allows then e.g. to build dialogue trees and recursively calculate probability of winning a dialogue, as described in [29, 5 Agent Strategy]. This approach is often referred to as *closed world assumption* which, generally speaking, means that the system has full knowledge of the world it reasons about.

This is of course a good start to work on argumentation strategies, but in a distributed environment it would mean that the first step of every dialogue would be to transfer the whole graph of arguments to it's participants which both can be time-consuming for large graphs and is inefficient, because eventually only a subset of it will be used in the dialogue. Additionally, it limits the system's ability to approximate the natural dialogue, because only in fairly simple cases humans are able to grasp and process the whole domain of dispute at once, especially at the beginning of the discussion.

With respect to above, the distributed AAC should rather lay on the *open world assumption*, which in the given case means that the we also allow situations where agents have only access to certain argumentation *subgraph* at any moment of the dialogue. Of course, their subgraphs must overlap in order to make any dialogue possible. For example, a referee (e.g. located at the server) could initiate the dialogue by sending each participant a certain subgraph of arguments, and than provide them with more knowledge as needed. The agents could also acquire the knowledge from each other. This automatically raises the issue of types of dialogue present in AAC, but we will leave that for a moment.

For now let us just emphasize that if agents have access only to a subgraph of arguments, than we need to decide how big can it be. Alternatively, we could decide how long can the agent "think" (analyze the argumentation graph, ask for more knowledge, etc.) before making a move in the dialogue. Which leads us to another basic issue. As one of the main purposes of AAC would be evaluation of argumentation strategies, the fact that agents participating in a discussion run on different, remote machines makes comparing strategies a more complex problem. Obviously, the faster the machine is the more agent's deliberation can take place in a given amount of time. So from the fact that certain agent won a dialogue (without specifying what exactly this means) does not immediately follow that it has a better strategy. One solution to this could be analogous to the one observed in combat sports, where typically competitors are divided into weight categories with one separate *opened* category where anyone can participate. So too, we could have in AAC different categories of participants depending on the strength of machines they run on. At this stage we are far from suggesting that this is the best solution, especially given the fact that the comparison of argumentation strategies can be based on much more factors than just the win-loss rules.

Apart from the issue of different agents' capabilities, evaluation of strategies requires also a good way of measuring the time a participant actually spends on "thinking" in opposition to the time spent on communication e.g. through the Internet. For the purpose of demonstration we can make a quick outline of some possible solutions to this problem:

- the simplest and least reliable: relaying on the agent and requiring him to sent the amount of time he spent on deliberation with his move;
- more resource-consuming: sending the agent an *immediate-response* message which it should send back with no delay — this would allow the referee to measure the time lost solely on communication;
- "lazy": assume that time constraints for agents' moves are so big, that communication delays do not matter.

Let us now return to the problem of knowledge distribution that emerged earlier in this section. Despite the fact that participants of AAC run in a distributed environment, they need to have the same understanding of argumentation framework instance (which we called an *argumentation graph*) that underlays their discussion.

One possible solution indicated few paragraphs earlier bases on an analogy to natural dialogues, which according to [24] can be divided into six categories: persuasion, inquiry, negotiation, information-seeking, deliberation and eristic. As Walton and Krabbe point out, it is not claimed that this is either complete or the only possible classification of dialogues, but it is a very convincing explanation of how the natural dialogue actually works and it certainly gives a hint on how to deal with the problem of knowledge distribution. The idea would be to introduce a second type of dialogue to the agents' interaction protocol, namely the information-seeking dialogue which (as Walton suggests in [23]) is engaged in when participants seek to acquire or give information.

At any given stage of a dialogue, it's participants can request to engage in information-seeking dialogue, either with the referee or with another participant, if the agent decides it has insufficient knowledge

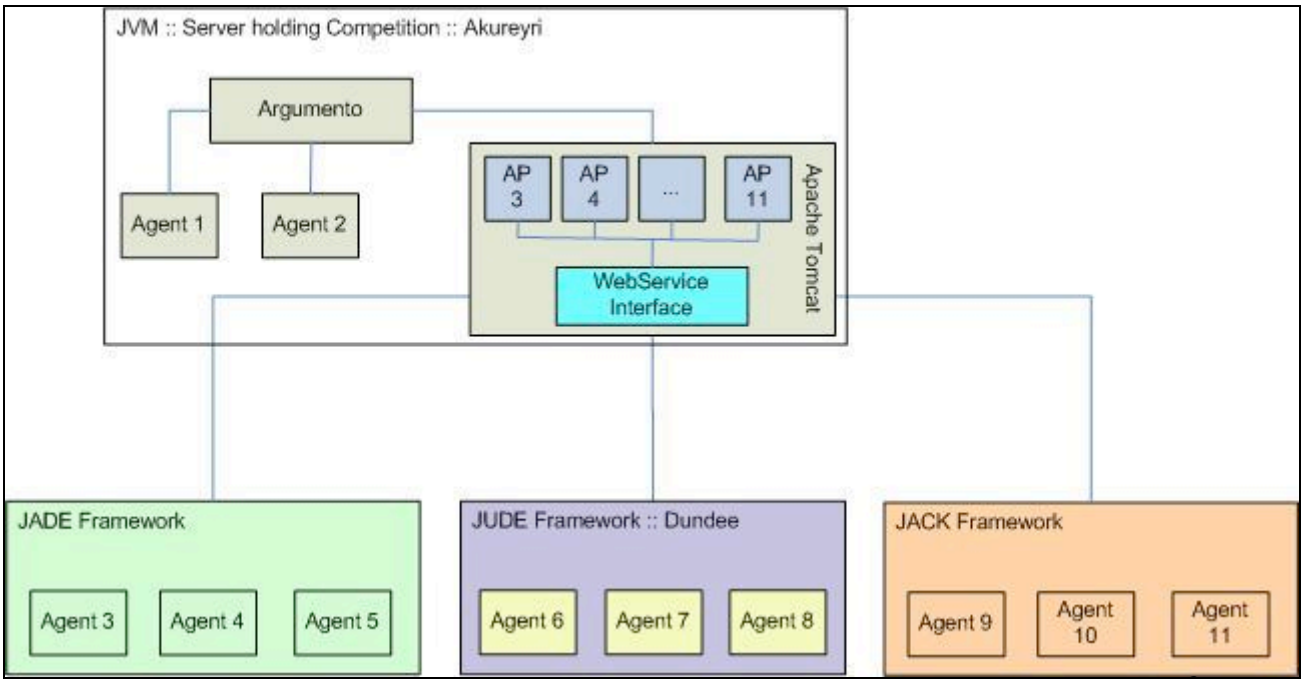


**Figure 1.** Proposed Distributed AAC Architecture

for continuing the discussion. As it was indicated earlier, we need to impose some constraints on the number of such inquiries in order to avoid blocking of the competition with long data transfers. Of course, if the information-seeking dialogue would be permitted among competition participants, they would be strongly required to obey the Gricean *Cooperative Principle* (vide [13]), more specifically the quality maxim that states:

"*Do not say what you believe to be false.*"

Or simply speaking: *Don't lie*. The application of this maxim as well as the whole Cooperative Principle should not be limited only to information-seeking dialogue, but in this case the absence of it most clearly defeats the point of having the dialogue.

*Argumento* is based upon Dung's argumentation framework as defined in [9]. This approach has many advantages, the most obvious being the simplicity of implementation, a small amount of information that needs to be contained in a single dialogue move and lack of commitment to any particular theory of internal argument structure. Nevertheless, the simplicity of this framework can also be a disadvantage in the sense that it gives a very limited amount of information to base argumentation strategy on: the binary attack relation.

Although extending *Argumento* with a more sophisticated argumentation framework is an attractive idea, we would not want to limit AAC's capabilities to just one solution knowing that there is a wide variety of proposals on how to define an argumentation framework. Hence, challenging but very interesting aim of the extension of *Argumento* to AAC would be to allow the evaluation of strategies based on different argumentation frameworks, whether it would be some developments of Dung's framework such as [6], a more elaborate framework like Vreeswijk's *Abstract Argumentation System* (vide [21]) or an approach based on a more conventional conception of argument's internal structure e.g. *Carneades Argumentation Framework* ([12]).

It makes sense, where possible, to build upon and exploit existing standards to facilitate the AAC. For example, utilizing agreed formats for argument interchange will lower the bar to entry for new AAC competitors. Whilst there are a variety of methods for representing and exchanging arguments, a nascent format, the Argument Interchange Format (AIF) [7] would be a useful tool that the AAC could exploit. The AIF enables storage and interchange of argumentation graphs in a distributed environment, but on the other it is not bound with any particular argumentation framework, and was in fact designed to support a range of differing argumentation frameworks.

Finally, we should consider the possibility of simulating the situation often found in natural argumentation, when it's participants do not share the same understanding of the domain of their dispute. Translating the problem to formal argumentation it would mean the possibility of having dialogues in which agents don't rely on identical graphs of arguments, although they should use the same argumentation framework. On the other hand one can imagine a situation where debating agents use different frameworks (e.g. bipolar and CAF) and try to somehow take these differences into account. A similar problem of merging different argumentation graphs was already addressed (in the context of Dung's AFs) in [8], where *merging operators* are introduced in order to produce a single graph. The difference of course is that in AAC we would need a more dialectical approach to the problem which remains to be developed.

To summarize, we have explored some of the implications of moving *Argumento* to a distributed environment. We are far from claiming completeness of this list, it is also not intended to propose ready solutions to given problems. Rather, it serves as a manifestation of the great potential that Arguing Agents Competition has and a speculation on possible future developments of this project.

## 5 DIALOGIC ARGUMENTATION ISSUES

Assuming the use of the AIF to represent monologic argument, it would be sensible to also adopt methods of dialogue representation,

that build upon the AIF, thereby enabling a uniform approach to dealing with arguments and the interactions that exploit them, in the AAC. One framework that has been proposed for representing the dialogical aspects of argumentation and which builds upon the AIF, is the AIF+ [19]. The AIF+ introduced the notion that movement from one locution to the next in a dialogue is licensed by a *Transitional Inference Scheme* which is analogous to the inference between propositions captured by argumentation schemes [22]. Using Transitional Inference Schemes, the AIF+ can be used to specify a dialogue protocol and to record the transcript of utterances made during a dialogue.

Given the flexibility of AIF for argument representation, enabling the interchange of arguments, and argument fragments, between the various agents and the *Argumento* engine regardless of underlying framework, it would seem prudent to utilize the AIF+ to interchange dialogue fragments, for example utterances within a given dialogue game, to represent the rules of the game, and to record the transcript of the final game state, so that the AAC can be represented at all levels within a consistent representational framework regardless of underlying semantics.

## 6 CONCLUSIONS & FUTURE WORK

This paper has examined the architecture of an AAC that would enable multiple, distributed, heterogeneous agents to engage in argument using an existing agent argumentation application. Work is currently ongoing to implement the web-service framework to enable the distribution of agents engaging in a competition so that they are not limited to running solely on the competition server. Once distribution is implemented the next step is to enable heterogeneous agents, implemented using a variety of agent frameworks, to join an *Argumento* competition proxied to instances of default *Argumento* agents to form the basic AAC implementation framework.

Future work will seek to expand on this basic platform by expanding the supported argumentation frameworks and game protocols. There are still a number of open questions about such an endeavor however, the most pressing concerns the competition protocols which must be developed so as not to prejudice any subsequent argumentation specific aspects. It should also be noted that in section 3 it is glibly stated that the "best" agent will be crowned champion, but the exact mechanism by which such a best agent will be determined is not yet apparent although there are some directions hinted at in [28] and [29].

In summation, the AAC is an exciting opportunity to build a new tool that researchers in argumentation can exploit, and that can attract new researchers to the area. It is also hoped that this paper will stimulate some discussion around the issue.

## ACKNOWLEDGEMENTS

We would like to acknowledge the implementational work on *Argumento* by Jenny Schulze and Tang Ming Yuan at the University of Akureyri, Iceland, as well as all of those other researchers currently building various aspects of the AAC.

## REFERENCES

[1] The Apache Software Foundation Apache Axis 2. http://ws.apache.org/axis2/, 2008.
[2] The Apache Software Foundation Apache Tomcat. http://tomcat.apache.org/, 2008.
[3] R. Arunachalam and N. Sadeh, 'The supply chain trading agent competition', *Electronic Commerce Research and Applications*, **4**, 63–81, (2005).
[4] J. L Austin, *How To Do Things With Words*, Oxford University Press, 1962.
[5] T. J. M. Bench-Capon, 'Specification and implementation of toulmin dialogue game', in *Proceedings of JURIX 98*, pp. 5–20, (1998).
[6] C. Cayrol, C. Devred, and M. Lagasquie-Schiex, 'Handling controversial arguments in bipolar argumentation systems', in *Computational models of argument (COMMA), Liverpool, 11/09/2006-12/09/2006*, eds., Paul E. Dunne and Trevor J.M. Bench-Capon, pp. 261–272, http://www.iospress.nl/, (septembre 2006). IOS Press.
[7] C. Chesnevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott, 'Towards an argument interchange format', *Knowledge Engineering Review*, **21**(4), 293–316, (2006).
[8] S. Coste-Marquisa, C. Devreda, S. Konieczny, M. Lagasquie-Schiexb, and P. Marquis, 'On the merging of dungs argumentation systems', *Artificial Intelligence*, **171**(10-15), 730–753, (2007).
[9] P. M. Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and *n*-person games', *Artificial Intelligence*, **77**, 321–357, (1995).
[10] Java Agent DEvelopment Framework. http://jade.tilab.com/, 2006.
[11] R. A. Girle, 'Knowledge organized and disorganized', *Proceedings of the 7th Florida Artificial Intelligence Research Symposium*, 198–203, (1994).
[12] T. F. Gordon and D. N. Walton, 'The carneades argumentation framework: Usuing presumptions and exceptions to model critical questions', in *Computational Models of Natural Argument, Proceedings of COMMA 2006*, eds., Poul E. Dunne and Trevor Bench-Capon, volume 144, pp. 195–207. COMMA, IOS Press, (2006).
[13] H. P. Grice, 'Logic and conversation', in *The Logic of Grammar*, eds., Donald Davidson and Gilbert Harman, 64–75, Dickenson Publishing Co., Encino, California, (1975).
[14] Agent Oriented Software Group. http://www.agent-software.com, 2006.
[15] C. L. Hamblin, *Fallacies*, Methuen and Co. Ltd, 1970.
[16] Calico Jack Ltd. http://www.calicojack.co.uk, 2005.
[17] P. McBurney and S. Parsons, 'Agent ludens: Games for agent dialogues', in *Game-Theoretic and Decision-Theoretic Agents (GTDT 2001): Proceedings of the 2001 AAAI Spring Symposium*, (2001).
[18] D. Moore and D. Hobbes, 'Computational uses of philosophical dialogue theories', *Informal Logic*, **18**(2 and 3), 131–163, (1996).
[19] C. Reed, S. Wells, G. W. A. Rowe, and J. Devereux, 'Aif+: Dialogue in the argument interchange format', in *2nd International Conference on Computational Models of Argument (COMMA 2008)*, (2008).
[20] J. R. Searle, *Speech Acts*, Cambridge University Press, 1969.
[21] G .A. Vreeswijk, 'Abstract argumentation systems', *Artificial Intelligence*, **90**(1–2), 225–279, (1997).
[22] D. N. Walton, *Argumentation Schemes for Presumptive Reasoning*, Lawrence Erlbaum Associates, 1996.
[23] D. N. Walton, 'The place of dialogue theory in logic, computer science and communication studies', *Synthese*, **123**, 327–346, (June 2000).
[24] D. N. Walton and E. C. W. Krabbe, *Commitment in Dialogue*, SUNY series in Logic and Language, State University of New York Press, 1995.
[25] M. P. Wellman, A. Greenwald, P. Stone, and P. R. Wurman, 'The trading agent competition', *Electronic Markets*, **13**(1), (2003).
[26] M. P. Wellman, P. R. Wurman, K. O'Malley, R. Bangera, S. Lin, D. Reeves, and W. E. Walsh, 'Designing the market game for a trading agent competition', *IEEE Internet Computing*, **5**(2), 43–51, (2001).
[27] M. Wooldridge, *An Introduction to Multiagent Systems*, John Wiley & Sons, Ltd, 2002.
[28] T. Yuan, J. Schulze, J. Devereux, and C. Reed, 'Towards an arguing agents competition: Building on argumento', in *CMNA 2008, Under Review*, (2008).
[29] T. Yuan, V. Svansson, D. Moore, and A. Grierson, 'A computer game for abstract argumentation', in *Proceedings of the 7th Workshop on Computational Models of Natural Argument (CMNA'07)*, (2007).