Formal Dialectic Specification

Simon Wells and Chris Reed

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

Abstract. Formal dialectic systems have been suggested as a means to model inter-agent communication in multi-agent systems. The formal dialectic systems of Hamblin are practical models for the computational implementation of such a system of argumentative dialogue.
This paper introduces a formal framework for the specification of Hamblin-type systems that has a range of benefits for theoretical work in the area including: yielding concise sets of clearly defined moves; allowing the moves of both existing and new games to be specified in a consistent manner; facilitating the use of dialectical shifts and dialogue embeddings independent of ruleset; facilitating the investigation of the coupling between sets of moves and dialogue situations; defining the attributes possessed by the general Hamblin-style formal dialectical system and thereby enabling the systematic exploration of the types of moves that these systems might encompass; and facilitating the rapid development of software applications that use formal dialectic to regulate communications.

1 Introduction

The formal dialectic systems after [1] have been proposed as a practical means to model the interactions between the participants in a dialogue. In computational environments, such dialectical systems have found application in various domains including the design, specification and implementation of conversation protocols in multi-agent systems. The dialogue in a formal dialectic system is conceived as a turn-taking game between two players whose utterances are constructed in terms of the types of moves allowed in the game. Games are traditionally specified in terms of sets of locutional rules, structural rules, commitment rules and, occasionally, completion rules. The locutional rules specify the possible moves that a player might make, the structural rules regulate which moves are allowed at a given point in the dialogue and the commitment rules govern how dialectical commitment is incurred or retracted as a result of the players moves.

The original Hamblin-style formal dialectic system, H in [1], was suggested primarily but not solely as a means to investigate some of the logical fallacies. As a practical investigational tool H was seen as an antidote to what Hamblin called the 'Standard treatment of fallacies' in which, he claimed, the common fallacies were listed without any deep examination of their underlying causes and the conditions from which they arose.

This dialectical approach to fallacy research was very productive yielding several families of games including those of Woods and Walton [2], Mackenzie [3] and Walton [4]. From fallacies the focus of dialectical systems research moved towards general theories of dialogical argumentation utilising commitment in the work of [5], commitment as a means to model belief in artificial intelligence in [6] and joint activities in [7].

Contemporary work with formal dialectical systems has concentrated on applying the theory to various aspects of computational systems. Mackenzie's game DC was used in [8] and [9] to research computer based learning systems. The application of formal dialectical systems to multi-agent systems is investigated in [10]. A theoretical basis for dialectical shifts within dialogues, the means to move from one dialogue type, as typified in [5], to another during a single dialogue is introduced in [11]. Some of the most recent work on argumentation in AI is presented in [12], which lays out new collaborative ventures not only in multi-agent systems, but also decision support, natural language, law, and knowledge representation.

2 Motivation

Formal-dialectic systems exhibit certain useful properties. These include explaining sequences of human utterances, a means to make decisions that can be less computationally intensive than game-theoretic based decision making [11] and allowing reasoning to occur in domains where knowledge is uncertain or limited.

Making use of a formal-dialectic game in a multi-agent system requires that a set of rules be devised and implemented as conversation protocols in agents. There are many games in the argumentation and multi-agent systems literature including [1], [3], [4], [5], [7], [6], [13], [14], [15], [16], [17], [18], that might make good candidates for protocols to regulate argumentative inter-agent communication. Unfortunately many of these have been suggested as means to tackle particular problems within argumentation, such as fallacious reasoning, rather than as general models of argumentative dialogue. Even where games have been formulated primarily as models of argumentative dialogue. it is difficult to efficiently compare the performance of the many different systems. Therefore it is not readily identifiable which games are better than others and under which circumstances and conditions this is so.

One way to determine which system of rules is most suitable is to compare them under a variety of conditions to determine particular properties that each set of rules might possess. Testing through implementation as suggested in [19] compliments our original intent of producing a computational implementation of a formal-dialectic system for inter-agent communication.

The problem with this though is that the games are specified in different ways, usually in natural language. This is sufficient to expose details of argumentative discourse for a theoretical audience but inadequate when the systems are implemented in software where ambiguity in the original specification can alter the overall behavior of the implemented system. A second problem is that the different specifications mean that multiple implementations are required, one for each formal-dialectic system which is both time and effort intensive. Each implementation would be limited to a single set of rules. This assumes that the set of rules selected are those most appropriate to governing inter-agent communication. A deeper concern with this approach is the lack of an examination of the underlying principles of a formal-dialectic system and the space of possible behaviors that such a system might exhibit.

A more efficient approach would be to draw out the common elements of the existing formal-dialectic systems and codify them in terms of a specification format that would enable a single computational implementation to afford the behaviour of any existing Hamblin-type formal-dialectic system. Some steps have been made towards a general framework for dialogue games, notably in [20] which suggests a theoretical framework but lacks the implementational detail and in [21] which looks at dialectical systems more generally and does not examine the entire family of games due to Hamblin. The aim of the current work is to develop not only a broad theoretical framework, but also an implementation that can act as a rapid application development tool for dialectical agents.

An initial approach would be to define the characteristics of the general formal-dialectic system and through examination of existing sets of rules determine the requirements for a framework and specification format in which the legal moves of any existing formal dialectic system can be specified. The formal dialectic systems, both those embodied in existing sets of rules and those suggested through examination of the properties of formal-dialectic systems in general, could then be compared and contrasted using a single framework which would eliminate behaviors introduced by differing implementations of the various systems. Ultimately this would allow an examination of the space of the possible sets of rules that might be embodied in a formal-dialectic system.

In order to allow the flexible definition, implementation, assessment of fit-for-purpose, and evaluation of dialogue systems, and classes of dialogue systems, what is required is a single, simple framework that can not only handle the full richness, diversity and expressive power of games described in the argumentation literature, but that can also form a direct bridge to rapid design and development of systems that implement those games in AI systems.

3 Towards A Framework for Game Specification

3.1 Hamblin's Formal-Dialectic and Successors

Hamblin conceived of the formal-dialectic as a means to investigate phenomena that occur in natural dialogue. This was achieved through the design of simple systems of precise rules which could be used to plot the properties of dialogue played out according to the rules. The formal dialectic can be generally defined thus;

> A regulated dialogue involving a number of participants who speak in turn in accordance with a set of rules that specify the form of what is said relative to the context and previous utterances in the dialogue.

On the surface this does not set the formal-dialectic of Hamblin much apart from the dialectic of [22] or the obligation game of medieval philosophy (see [1] chapter 8) but Hamblin further specifies some elements of the possible systems. The number of participants is set at two, the dialogue context refers to the specification of an interaction between the players and the previous occurences refer either to previous locutional acts within the dialogue or to the contents of the players commitment stores.

Practically this requires that the rules of a formal-dialectic system specify contexts of interaction, for example, that move X_1 must follow move X_0. Further a rule may prescribe that a players commitment store maintain a particular commitment for that interaction to be legal, or that a specific move has been made at an earlier point in the dialogue. A system of formal-dialectic therefore has three core characteristics which must be reflected in a formalisation of that system:

1. Interaction contexts
2. Commitment store contents
3. Dialogue history

Hamblin's game, H, specifies a set of moves that embody these characteristics. The rules of H are presented in two categories, those that describe the syntactical structure of the dialogue, and those that operate on a players commitment stores. Successive games in [4], [5], [3] and in [7], reutilse the format of H with additional categories of rules such as the locutional rules that specify what can be said, the structural rules which are the syntactical rules renamed, and the closure rules, a seldom used group of rules that suggest a means to determine when a dialogue is at an end.

3.2 Specifying Sets of Moves

Common to the formal dialectic systems of [1], [3], [4], [5] and [7] is the notion of the legal move. The only way to alter the state of the dialogue is to make a legal move specified in terms of a locution and usually some propositional content. Usually the order in which moves can be made depends heavily upon which move was made in the previous players turn. This is the primary mechanism that allows the dialectical systems to regulate the flow of conversation and thereby allow the conversants to engage in realistic dialogues by restricting which moves are valid at any given point. In practice, only a subset of the possible moves is available for the current players to use depending upon earlier moves.

Hamblin introduces the notion of commitment as a means to maintain consistency in a player's utterances. Through making a move in a formal-dialectic system a player can incur some form of propositional commitment. The commitments incurred are added to a public record of all commitments incurred in the dialogue by each player, this record is called the commitment store. Hamblin suggested that it be required of each new commitment that it can be added without inconsistency to the incurring player's store. The notion of commitment and stores of incurred commitments has been applied to multi-agent systems research in [23], [24] [25] and [26]. [27] presents the DIAGAL agent communication language and a series of games that use player incurred commitment to regulate flexible conversation policies.

Each player maintains a set of commitment stores. There are two basic types of commitment store, the static commitment store and the dynamic commitment store. The static commitment store, also called the veiled [4] or dark-side [5] commitment-store is not altered by moves in the dialogue and it's contents are fixed before the dialogue commences. The dynamic commitment store [1] is a set of propsitional commitments that alter depending upon the moves the players make in the dialogue. The dynamic commitment stores are indexed by turn so that each player's set of commitments at each turn in the dialogue can be examined.

An examination of the existing rules yields a set of parameters that govern whether a particular move is admissible, and a set of effects that take place as a result of the move being made. If the specification of individual moves is made in terms of their legality requirements and effects, then the separate specification of commitment, structural and locutional rules is not required. The legality requirements and effects can be structured in terms of the pre-conditions and post-conditions required by the move. The pre-conditions state the conditions that must hold for the move to be legally made, for example, the contents of the players commitment stores and the previous moves made in the dialogue. The post-conditions specify the alterations that should be made to the players commitment stores as a result of the move and the set of moves that is allowed in response to the current move. This specification is therefore reminiscent of a planning style approach to dialogue, such as was developed in early work by [28].

Specifying Individual Moves An individual move can be modelled thus;

Move *Name of move*
Pre-Conditions *Conditions that must hold for the move to be legally made*
Post-Conditions *Conditions that must hold for the move to be satisfactorily completed*

In order to specify the pre-conditions and post-conditions that an individual move may possess, some game variables must be defined;

A game comprises a set of players, Π={P,O} for the proponent and opponent, a set of propositions Λ (of arbitrary atomic tokens), a set of locutions Φ. A given dialogue is a sequence of turns, that can be constructed as a relation R mapping $\mathbb{N} \rightarrow T$, where T are possible turns, i.e. T: Φ x Λ x Π . For syntactic convenience T_n refers to the triple $\langle \Phi, \Lambda, \Pi \rangle$ identified by R(n).

The player's commitments are recorded in commitment stores designated $C\pi_n$ where $\pi \in \Pi$ identified here for an arbitrary agent S, and form a sequence of sets of commitments. The player's may maintain any number of individually labelled commitment stores as set out in the rules for the game being played.

There are several general classes and subclasses of both pre- and post-conditions. These are outlined below with an explanation of each.

Pre-Conditions

1. Commitment Store Contents
   (a) $C \in CS_n$
   Commitment C is currently in commitment store CS
   (b) $C \notin CS_n$
   Commitment C is not currently in commitment store CS
   (c) $\exists m$, $m \in \mathbb{N}.C \in CS_{n-m}$
   Commitment store CS has previously contained commitment C
   (d) $\exists m$, $m \in \mathbb{N}.C \notin CS_{tn-m}$
   Commitment store CS has not previously contained commitment C
2. Previous Moves
   (a) $T_{n-1} = \langle \Phi, \Lambda, \Pi \rangle$
   The immediate previous turn comprised locution Φ made with content Λ by player Π
   (b) $T_{n-1} \neq \langle \Phi, \Lambda, \Pi \rangle$
   The immediate previous turn did not comprise locution Φ made with content Λ by player Π
   (c) $\exists m$, $m \in \mathbb{N}$, m<n.$T_{n-m} = \langle \Phi, \Lambda, \Pi \rangle$
   A previous turn comprised locution Φ made with content Λ by player Π
   (d) $\forall m$, $m \in \mathbb{N}$, m<n.$T_{n-m} \neq \langle \Phi, \Lambda, \Pi \rangle$
   A previous turn did not comprise locution Φ made with content Λ by player Π

Post-Conditions

1. Alterations to Commitment Stores
   (a) $\mathrm{CS}_{n+1} = \mathrm{CS}_n \cup \{\mathrm{C}\}$
   Commitment C is added to commitment store CS
   (b) $\mathrm{CS}_{n+1} = \mathrm{CS}_n \setminus \{\mathrm{C}\}$
   Commitment C is removed from commitment store CS
2. Legal Responsive Moves
   (a) $\mathrm{T}_{n+1}{=}\langle \Phi,\ \Lambda,\ \Pi\rangle$
   The next turn must comprise locution Φ made with content Λ by player Π
   (b) $\mathrm{T}_{n+1}{\neq}\langle \Phi,\ \Lambda,\ \Pi\rangle$
   The next turn must not comprise locution Φ made with content Λ by player Π
   (c) $\exists m,\ m{\in}\mathbb{N}.\mathrm{T}_{n+m}{=}\langle \Phi,\ \Lambda,\ \Pi\rangle$
   A future turn must comprise locution Φ made with content Λ by player Π
   (d) $\forall m{,}m{\in}\mathbb{N}.\mathrm{T}_{n+m}{\neq}\langle \Phi,\ \Lambda,\ \Pi\rangle$
   A future turn must not comprise locution Φ made with content Λ by player Π

Due to the manner in which each turn is constructed, any combination of move, propositional content and player can be specified as a past or future condition of a move. This is because a fully formed legal move is made by a player and comprises a locution and some propositional content. This means that the space of possible moves in a Hamblin-style formal dialectical system can be delineated through application of the dialogue variables.

By utilising this format, the number of rules required to specify a set of moves for a dialectical system is reduced merely to the number of legal moves in the system. For example, the legal moves of DC can be specified as follows: N.B. The proponent is denoted P and the opponent O. The proponent's commitment store is denoted CP and the opponent's commitment store, CO.

1. **Move** Statement(S_x)
   Pre ∅
   Post $\mathrm{CP}_{n+1} = \mathrm{CP}_n \cup \{\mathrm{S}_x\}$
   $\wedge\ \mathrm{CO}_{n+1} = \mathrm{CO}_n \cup \{\mathrm{S}_x\}$
2. **Move** Denial(S_x)
   Pre ∅
   Post $\mathrm{CP}_{n+1} = \mathrm{CP}_n \cup \{\neg\mathrm{S}_x\}$
   $\wedge\ \mathrm{CO}_{n+1} = \mathrm{CO}_n \cup \{\neg\mathrm{S}_x\}$
3. **Move** Defense(S_y)
   Pre ∅
   Post $\mathrm{CP}_{n+1} = \mathrm{CP}_n \cup \{\mathrm{S}_y\}$
   $\wedge\ \mathrm{CP}_{n+1} = \mathrm{CP}_n \cup \{\mathrm{S}_y{\rightarrow}\mathrm{S}_x\}$
   $\wedge\ \mathrm{CO}_{n+1} = \mathrm{CO}_n \cup \{\mathrm{S}_y\}$
   $\wedge\ \mathrm{CO}_{n+1} = \mathrm{CO}_n \cup \{\mathrm{S}_y{\rightarrow}\mathrm{S}_x\}$

4. **Move** Withdrawal(S_x)
   Pre $\emptyset$
   Post $CP_{n+1} = CP_n \setminus \{S_x\}$
5. **Move** Challenge(S_x)
   Pre $\emptyset$
   Post $CP_{n+1} = CP_n \setminus \{S_x\}$
   $\wedge\ CP_{n+1} = CP_n \cup \{\mathrm{Why}S_x?\}$
   $\wedge\ CO_{n+1} = CO_n \cup \{S_x\}$
   $\wedge$ ($T_{n+1}=\langle \text{Defense}, S_x, O\rangle$
   $\vee\ T_{n+1}=\langle \text{Resolve}, S_x, O\rangle$
   $\vee\ T_{n+1}=\langle \text{Withdrawal}, S_x, O\rangle$)
6. **Move** Question(S)
   Pre $\emptyset$
   Post $T_{n+1}=\langle \text{Denial}, S_x, O\rangle$
   $\vee\ T_{n+1}=\langle \text{Statement}, S_x, O\rangle$
   $\vee\ T_{n+1}=\langle \text{Withdrawal}, S_x, O\rangle$
7. **Move** Resolve(S)
   Pre $\emptyset$
   Post $T_{n+1}=\langle \text{Statement}, S_x, O\rangle$
   $\vee\ T_{n+1}=\langle \text{Withdrawal}, S_x, O\rangle$

Through using this framework the game DC can be modified easily to yield the variant game DD [3] by replacing the post-conditions of the Statement(S_x) move with the following:

Post $CS_{n+1} = CP_n \cup \{S_x\}$
$\wedge\ CS_{n+1} = CP_n \setminus \{\mathrm{Why}S_x?\}$
$\wedge\ CH_{n+1} = CO_n \cup \{S_x\}$

3.3 Specifying Dialogue Stages

The use of dialogue stages allows the practical implementation of dialectical shifts, the movement from one type of dialogue to another [5], and dialogue embeddings [11], the functional encapsulation of an entire child dialogue within a parent dialogue. [29] identifies four stages that dialogue might proceed through including the confrontation stage, opening stage, argumentation stage and the concluding stage.

A practical example of the use of stages can be seen in the specification of the game PPD_0 in [5]. In PPD_0 the moves allowed in the opening stage of a dialogue are restricted and this situation continues until the contents of the players commitment stores match a particular set of conditions. This is an example of a dialogue shift because the dialogue moves from an expository situation in which the conversants are laying out their initial commitments to a more structured dialogue in which the conversants may explore each others arguments. This is not exactly the same as a dialogue shift though, the dialogue type according to the Walton and Krabbe typology [5] remains the same, persuasion dialogue, but the persuasion dialogue may move through several stages before completion.

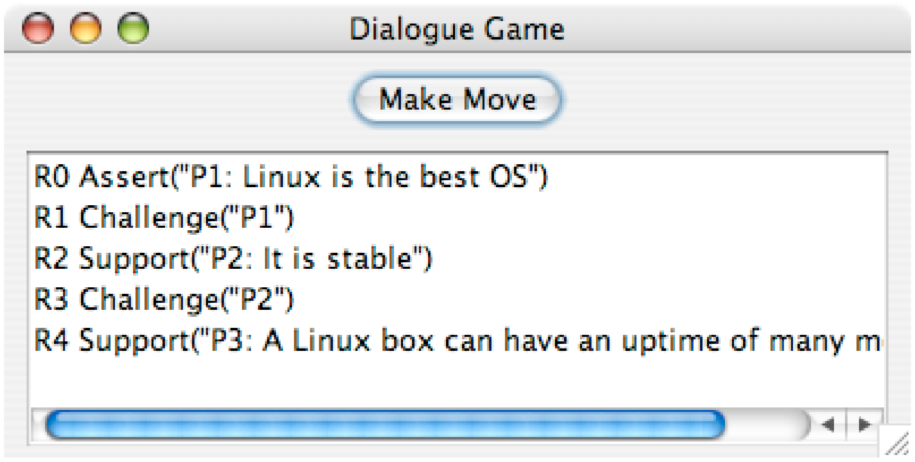
A practical means to implement this type of shift is to bind each set of moves into a stage which specifies a set of entry and exit conditions. In single stage games such as H, DC et al where the set of rules is fixed throughout the entire dialogue, the entry-conditions set out the point of issue between the conversants and the exit-conditions can be used to specify when the dialogue is complete, this allows games to make use of win-loss completion conditions that were not specified in the original rules. In PPD_0 this is sufficient to allow an opening stage in which the players commitment stores are set up and the legal moves are restricted. When the exit-conditions of the opening stage of PPD_0 are met the entry-conditions for the next stage, the persuasive dialogue stage, are examined and if met the dialogue proper begins. The exit conditions of the persuasive dialogue stage set out the win/loss conditions of the dialogue as a whole. This mechanism allows the dialogue to proceed in a series of discrete stages with a complete set of rules defined and tailored to suit the requirements of each stage.

A stage therefore consists of a set of permissible moves and a set of conditions under which the stage can be opened and closed. An effect of this formulation is that a dialogue-stage can be set up to embed entirely within another dialogue-stage through a careful formulation of the rules governing the entry and exit to the embedded sub-stage. As a result the stages that a dialogue might move through need not be linear. Dependent upon earlier occurrences in the dialogue and the formulation of entry- and exit-conditions, differing paths through the graph of stages might arise.

3.4 Implementation

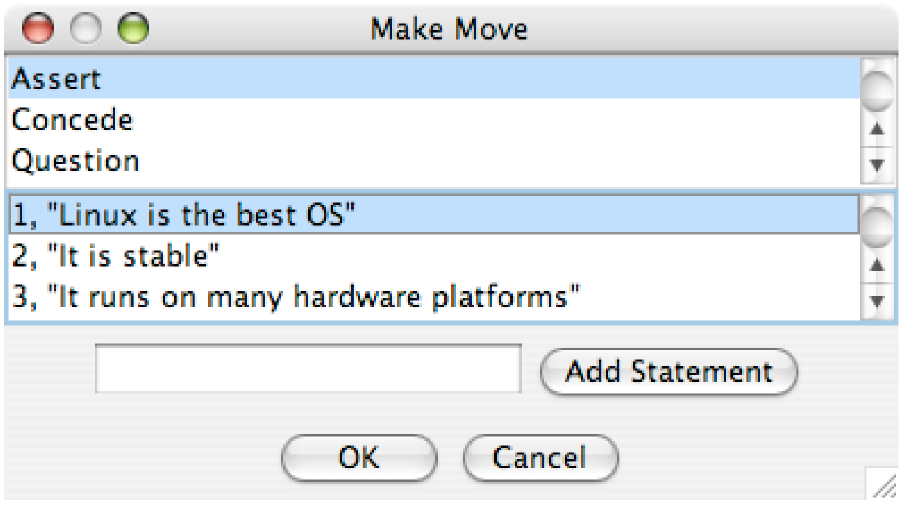
The current implementation of the framework can be seen in figures 1 and 2. It is written in Java and makes use of XML to specify the sets of legal moves that constitute a formal-dialectic system. The use of XML allows sets of rules to be swapped between conversants and means that some degree of run-time code generation can occur. This means that the dialectic system governing a dialogue is not hard-coded but flexible, allowing sets of rules to be tailored towards specific dialogue contexts, whether the dialogue types in [5] or those pertaining to social or cultural situations. This avoids the situation found in [8] and [9] in which a lot of work is expended on implementing and refining a single formal-dialectic system.

Development is now under way to refine the implementation in the form of a module for the JackDaw agent framework [30] a lightweight, flexible, industrial-strength agent platform. JackDaw agents make use of software modules to provide added functionality to the core agent capabilities. This development work will set the formal-dialectic framework firmly within the scope of multi-agent systems allowing agents that embody the resulting module to engage argumentative dialogue governed by formal-dialectic.



R0 Assert("P1: Linux is the best OS")
R1 Challenge("P1")
R2 Support("P2: It is stable")
R3 Challenge("P2")
R4 Support("P3: A Linux box can have an uptime of many m

Fig. 1. The main dialogue window showing a dialogue in progress



1, "Linux is the best OS"
2, "It is stable"
3, "It runs on many hardware platforms"

Fig. 2. The make move dialogue

By utilising a specification framework the rapid development, implementation and evaluation of Hamblin-type formal-dialectic systems is possible without the need to develop an individual implementation for each game.

4 The Benefits

The immediate benefits of this are that a single implementation can be constructed that uses concisely written, exactly specified and consistent rulesets that may be altered as required at design-time or run-time to suit the current context of dialogue. The implementation of dialogue shifts and embeddings is catered for through the use of dialogue stages which can be applied to any ruleset regardless of whether the original formulation of rules supported them.

The computational implementation of dialectical systems, especially if they are to communicate with other systems utilising the same set of rules, require that the results of each individual move be explicitly specified. Otherwise each implementation is subject to the system designer's interpretation of the rules which can lead to differences between the way systems would react given identical circumstances. Use of the suggested framework for the specification of formal-dialectic systems means that all existing systems can be written using a single nomenclature that exactly specifies what the effect of each variable should be. Any existing game can be specified merely by selecting the correct set of moves and specifying the appropriate pre-conditions and post-conditions for those moves. Related games can then be created simply by changing an individual rule or a component of that rule.

From here the space of possible rulesets for Hamblin-style formal dialectic systems may be examined. If a particular dialogue type is embodied in a particular set of rules governing how the dialogue should proceed then dialectical shifts and embeddings can be implemented as a means to shift between those types merely by altering the rules to that required by the particular dialogue type during run-time and by agreement between the conversants. The generic framework allows varying sets of rules to be compared and evaluated as to their applicability to particular dialectical situations. This continues the work outlined in [5] which suggests that new rules for PPD_0 type dialogues be formulated as required. It also extends earlier work by allowing, for example, games such as DC to make use of win/loss conditions to determine the result of a dialogue. A survey of dialogue situations and the rules applicable therein leads to a tighter coupling of rules to situation and facilitates improvements in the expressive capabilities of implemented systems.

A partial typology of dialogue types is established in [5]. Dialogue types are characterised by their initial situation, overall goal and individual participants aims. Sets of rules should be tailored to particular instances of a dialogue type.

Further distinctions might be drawn between rulesets applicable in differing social situations. For example, in a legal setting the rules by which a dialogue might proceed are very different from those that obtain at a hotel bar.

One set of rules will not suffice to govern all of the types of possible dialogues. When an agent uses a single agent-communication language (ACL) to govern all of its communicative interactions, the primitives of the ACL will necessarily be of the lowest common denominator. They will be applicable to as many situations as possible but not well suited to providing the best set of primitives for every dialogue context [31]. Analogously, were one set of rules proposed to govern all types of dialogue it could only service the common elements of each dialogue and would not necessarily embody the best set of rules for that situation. As the dialogue changes so the rules should change to suit. This is dealt with implicitly in the game PPD_0, where the opening stage shifts to the persuasive stage in response to the changing factors of the dialogue, with a resulting change in the set of legal moves. This concept could be extended to meet the needs of a diverse set of dialogue situations where the set of legal moves would be kept fluid depending upon the character of the dialogue currently underway. As the dialectical situation alters with the flow of dialogue, so the rulesets pertaining to the current dialectical situation should also change. A framework for formal dialectical systems that allows rulesets to be loaded at runtime gives formal-dialectic increased flexibility to handle a wide range of communicative situations, broadening the scope of computational implementations and their ability competency during differing dialogues as the goals and situations of the participants change.

5 Conclusions

The framework outlined herein sets out a means to utilise both existing formal dialectic systems after [1] and new formulations of rules within a single computational implementation. Where formal dialectic systems have been used as conversation protocols to govern the interactions between systems, there is now a common format within which sets of rules can formulated and exchanged. This allows researchers to easily explore new sets of rules within the context of a Hamblin-type formal dialectic system and it allows systems to swap sets of rules at runtime so that the conversation protocol governing the systems interactions can be tailored to the situation, dialogical or social, that it is engaged in.

Provision is made for existing games to take advantage of newer developments in argumentation research such as the dialogue stages of [5] and the dialogue embeddings of [11]. This not only extends the scope of many of the existing systems, including H, DC and CB, but also any Hamblin-type formal-dialectic system to make use of many sets of rules, each tightly coupled to a particular dialogue situation or context, and to move between those rulesets as needed.

The framework and associated implementation provide a means for multi-agent systems to make practical use of the rich, diverse and expressive systems presented in the argumentation literature. A single, simple, consistent approach is provided toward the specification, implementation and evaluation of dialogue systems that can meet the growing demands of sophisticated agents in complex domains.

6 Acknowledgements

This research is funded by EPSRC under the Information Exchange project (GR/S03706/01). Gratitude is expressed to Calico Jack Ltd. for their JackDaw agent framework and JUDE development environment.

References

1. Hamblin, C.L.: Fallacies. Methuen and Co. Ltd. (1970)
2. Woods, J., Walton, D.N.: Arresting circles in formal dialogues. Journal Of Philosophical Logic **7** (1978) 73–90
3. Mackenzie, J.D.: Question begging in non-cumulative systems. Journal Of Philosophical Logic **8** (1979) 117–133
4. Walton, D.N.: Logical Dialogue-Games And Fallacies. University Press Of America (1984)
5. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue. SUNY series in Logic and Language. State University of New York Press (1995)
6. Girle, R.A.: Knowledge organized and disorganized. Proceedings of the 7th Florida Artificial Interlligence Research Symposium by the Florida AI Research Society (1994)
7. Girle, R.A.: Commands in dialogue logic. Practical Reasoning: International Conference on Formal and Applied Practical Reasoning, Springer Lecture Notes in AI (1996)
8. Moore, D., Hobbes, D.: Computational uses of philosophical dialogue theories. Informal Logic **18** (1996) 131–163
9. Yuan, T., Moore, D., Grierson, A.: A conversational agent system as a test-bed to study the philosophical model dc. In: 3rd Workshop on Computational Models of Natural Argument (CMNA'03). (2003)
10. McBurney, P., Parsons, S.: Agent ludens: Games for agent dialogues. In: Game-Theoretic and Decision-Theoretic Agents (GTDT 2001): Proceedings of the 2001 AAAI Spring Symposium. (2001)
11. Reed, C.: Dialogue frames in agent communication. In: Proceedings of the 3rd International Conference on Multi Agent Systems, IEEE Press (1998)
12. Reed, C., Norman, T.: Argumentation Machines. Kluwer Academic Publishers (2003)
13. Amgoud, L., Parsons, S.: Agent dialogues with conflicting preferences. Pre-Proceedings of the Eighth International Workshop on Agent Theories (2001)
14. Amgoud, L., Parsons, S., Maudet, N.: Arguments, dialogue, and negotiation. Proceedings of the Fourteenth European Conference on Artificial Intelligence (2000)

15. Dignum, F., Dunin-Keplicz, B., Verbrugge, R.: Agent theory for team formation by dialogue. Intelligent Agents VII: Proceedings of the Seventh International Workshop on Agent Theories, Architectures and Languages (ATAL 2000) (2000)
16. Dignum, F., Dunin-Keplicz, B., Verbrugge, R.: Creating collective intention through dialogue. Logic Journal of the IGPL (2001)
17. Hitchcock, D., McBurney, P., Parsons, S.: A framework for deliberation dialogues. Proceedings of the Fourth Biennial Conference of the Ontario Society for the Study of Argumentation (2001)
18. McBurney, P., van Eijk, R.M., Parsons, S., Amgoud, L.: A dialogue-game protocol for agent purchase negotiations. Journal of Autonomous Agents and Multi-Agent Systems **7** (2001) 235–273
19. Maudet, N., Moore, D.: Dialogue games as dialogue models for interacting with, and via, computers. Informal Logic **3** (2001)
20. Maudet, N., Evrard, F.: A generic framework for dialogue game implementation. In: roceedings of the Second Workshop on Formal Semantics and Pragmatics of Dialog. (1998)
21. Bench-Capon, T.J.M., Geldard, T., Leng, P.H.: A method for the computational modelling of dialectical argument with dialogue games. Artificial Intelligence and Law **8** (2000) 233–354
22. Rescher, N.: Dialectics, "A Controversy-Oriented Approach to the Theory of Knowledge. State University of New York Press, Albany. (1977)
23. Singh, M.P.: Multi-agent systems as spheres of commitment. International Conference on Multi-Agent Systems (ICMAS) (1996)
24. Singh, M.P.: Commitments among autonomous agents in information-rich environments. 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW) (1997)
25. Singh, M.P.: An ontology for commitments in multi-agent systems. Artificial Intelligence and Law (1998)
26. McBurney, P., Parsons, S.: The posit spaces protocol for multi-agent negotiation. Advances in Agent Communication (2004)
27. Maudet, N., Chaib-draa, B., Labrie, M.A.: Request for action reconsidered as dialogue game based on commitments. Workshop on Agent Communication Language (AAMAS'02) (2002)
28. Cohen, P.R., Perrault, R.: Elements of a plan-based theory of speech acts. Cognitive Science **3** (1979) 177–212
29. van Eemeren, F.H., Grootendorst, R.: Argumentation, Communication, and Fallacies. A Pragma Dialectical Perspective. Lawrence Erlbaum Associates (1992)
30. http://www.calicojack.co.uk: Calico jack website (2004)
31. Reed, C., Norman, T., Jennings, N.: Negotiating the semantics of agent communication languages. Computational Intelligence (2002)