Development and Evaluation of a System for Educational Debate

Tangming Yuan[1] and **David Moore**[2] and **Alec Grierson**[3]

Abstract: This paper reports research concerning issues involved in adopting a computational dialectics approach to develop a computerised system for educational debate. In particular, we consider the construction and usability evaluation of a human computer debating system. The system has been fully implemented and an initial usability evaluation has been undertaken, from which the results are essentially favourable. It is anticipated that this work will contribute to the development of human-computer dialogue in general and of computer-based educational dialogue in particular.

Keywords: computational dialectics; human-computer debate; dialogue model; computational strategic heuristics; usability

1. INTRODUCTION

This paper reports our research involved in adopting a computational dialectics approach to the development of a human computer debating system. The system is expected to be used to educational advantage - to develop students' debating and reasoning skills and domain knowledge (a similar educational argument for a dialectical approach is made by Ravenscroft and Matheson (2002) and Moore (2000)). A fundamental element underlying the proposed computer debate system is the dialogue model. A previous paper (cf. Yuan et al. 2003) considered the influential dialogue model "DC" (Mackenzie 1979), and developed a further system "DE". The motivation behind this development is that the underlying dialogue model of the debating system is required to have the ability to pick out fallacious argument and common errors when they occur during the course of debate. DE was evaluated using conversational simulations in which two computer agents debated with each other using the DE model. The result shows improvement over DC in preventing fallacious arguments and common errors. In particular, DE appears advantageous over DC in preventing the fallacy of question begging, inappropriate challenges and the straw man fallacy, and appropriate handling of the issue of repetition of moves.

The "DE" model makes five move types available to both participants in the dialogue: assertions (P, Q, etc. or the truth-functional compounds), questions ("Is it the case that P?"), challenges ("Why P?"), withdrawals (e.g. "no commitment P") and resolution demands ("resolve whether P"). There are five rules regulating commitment stores of both participants: the initial commitment of each participant is null; after the withdrawal of P, the statement P is not included in the move maker's store; after a statement P, unless the preceding event was a challenge, P is included in the move maker's assertion list and the dialogue partner's concession list, and 'Not P' will be removed from the move maker's concession list if it is there; after a statement P, if the preceding event was "Why Q?", "P" and "If P then Q" are included in the move maker's assertion list and the dialogue partner's concession list, and 'Not P' and 'Not (If P then Q)'are removed from the move maker's concession list if they are there; a challenge of P results in P being removed from the store of the move maker if it is there.

In addition to these commitment rules, participants in a dialogue using the DE model are required to adopt seven dialogue rules: (i) participants may make one of the permitted types of move in turn; (ii) mutual commitment may not be asserted unless to answer a question or a challenge; (iii) the question "P?" may be answered only by P, "Not P" or "no commitment P"; (iv) "Why P?" must be responded to by a withdrawal of P, a statement acceptable to the challenger, or a resolution demand of any of the commitments of the challenger which immediately imply P. A statement S is "acceptable" to participant A at a stage n, just in case S is at stage n a commitment or a de facto commitment or a new commitment of A's store; (v) resolution demands may be made only if the dialogue partner has an immediately inconsistent conjunction of statements, or withdraws or challenges an immediate consequent of his commitments; (vi) a resolution demand must be followed by withdrawal of one of the offending conjuncts, or affirmation of the disputed consequent; (vii) "Why P?" may not be used unless P is on the assertion list of the dialogue partner.

A particular concern with DE, especially from a computational perspective, is that it leaves much to the discretion of the user of the model. For example, after a challenge (why P?), various options are open: one can respond with no commitment to P, or a resolution demand (in some circumstances) or a ground for P. Further, there is no guidance within the rules as to the content of the ground. Similarly, after a withdrawal or a statement, there are no restrictions on the move types or move contents. All DE does is to legitimise a set of move types given the prevailing circumstances, and occasionally give some indication of the semantic possibilities. In a human-computer debate setting, it is crucial therefore that the computer is given some means of selecting between available possibilities, e.g. to maintain

[1] Faculty of Information Technology, University of Akureyri, Iceland, email: yuan@unak.is

[2] School of computing, Leeds Metropolitan University, United Kingdom, email: d.moore@leedsmet.ac.uk

[3] School of computing, Leeds Metropolitan University United Kingdom, email: a.j.grierson@leedsmet.ac.uk

focus after a statement or a withdrawal, so that the produced moves are appropriate at the pragmatic level. This choice must be based on some suitable strategy (Maudet and Moore 2001). Appropriate strategic knowledge is, then, essential if the computer is to produce high quality dialogue contributions. Given this, a set of strategic heuristics for the computer to adopt to enable it to function as a dialogue participant has been proposed (Yuan 2004). The system is currently configured as what can be described as a "partially honest" agent in that the computer is allowed to insist on its own view for the sake of argument even though it may have more reasons in favour of the user's view. The debate strategies for such an agent have been developed on the basis of experimental studies of people using the DC framework, and posits three levels of decisions to consider (cf. Moore 1993). At level 1 the issue is whether to retain the focus or to change it. "Retain the focus" can be taken to mean continuing the same line of argument, which will involve continuing to attempt to substantiate or undermine a particular proposition. At level 2 there are again two alternatives for the computer. It can seek to demolish the user's position, that is seek to have him remove from his Commitment Store (CS) propositions which he has used to support his thesis. The ultimate aim is to remove all support of the user's position. Alternatively, the computer can build its own position, by making statements the acceptance of which, or asking questions the answers to which, ultimately imply the truth of its thesis. At level 3 the decision involves which method to adopt in fulfilment of the objective set at levels 1 and 2. Detailed heuristics for level 3 can be seen in (Yuan 2004). Level (1) and (2) refer to strategies which apply only when the computer is facing a statement or withdrawal, while level (3) applies in every DE game situation. In the current debating system, the computer checks the level 3 methods first; if there are level (3) methods available, the level (1) and (2) decisions will be automatically applied. However, if there is no level (3) method available, level (1) and (2) decisions will come into play.

In this paper, we seek to further the investigation of the debating system in several steps. Firstly, we discuss the construction of a human-computer debate prototype operationalising the dialogue model and strategy outlined above; the system asks the user their opinion on a controversial issue, adopts the opposite position and engages the user in debate on the issue. We then discuss our preliminary usability evaluation of the system. Finally, we discuss our intended future work concerning the development of the human-computer debating system.

2. A HUMAN-COMPUTER DEBATE PROTOTYPE

The approach, then, is to use the DE dialogue model and the dialogue strategy outlined above as the basis for a human-computer debate system. A fully functional prototype (currently able to debate the issue of capital punishment) has been built by the authors, using the Java programming language. The system enables the user to conduct a debate

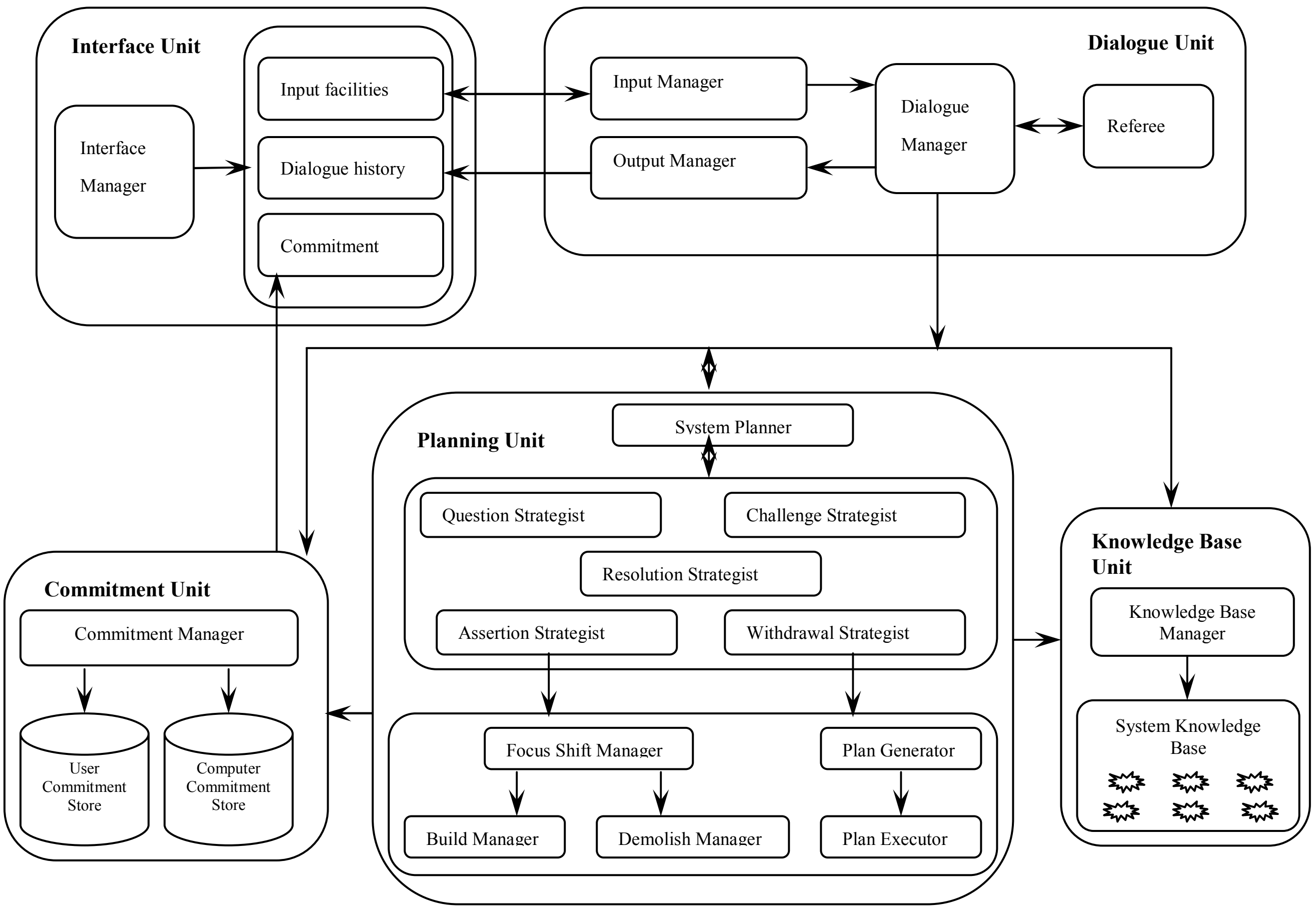

Figure 1 System Architecture

with it on the controversial issue of capital punishment. The computer can adopt either a proponent or an opponent role. That is, if the user chooses to support the view of “capital punishment is acceptable”, the computer will adopt the opposite view “capital punishment is not acceptable”, and vice versa. The system then engages the user in debate on the topic of capital punishment, given these initial positions on the issue. The system architecture is shown in figure 1. This remainder of this section provides details of the main components, input facilities and knowledge representation of the system.

2.1 System Components

There are five main units of the system: the interface unit, the dialogue unit, the commitment unit, the planning unit and the knowledge base unit. The *interface unit* provides the system’s user interface (see figure 2). It provides a dialogue history, which records the debate, and commitment-stores to show both the user’s and the computer’s commitment store contents. Input facilities enable the user to select the move type and move content from a menu. The interface manager enables the user to save the debate history and to change the background colour of the interface, and provides help facilities.

The *dialogue unit* can be regarded as the dispatch centre of the dialogue interactions. This unit consists of an input manager, a dialogue manager, a referee and an output manager. The input manager provides dynamic support for the user’s input, in that it makes available to the user only those move types permissible under the DE rules given the prevailing state of the dialogue. It then delivers the user input to the dialogue manager. The dialogue manager controls the turn taking of the interaction and is in charge of the input manager, output manager, referee, the commitment unit and the planning unit. Each user’s move will be passed to the DE referee for judgement. If the move is legal, the commitment manager will be called to update the commitment stores, the output manager will be called to update the dialogue history and the planning unit will be called to make a move on behalf of the computer. In the event of an illegal move, the referee will post a message and request the user to make another move.

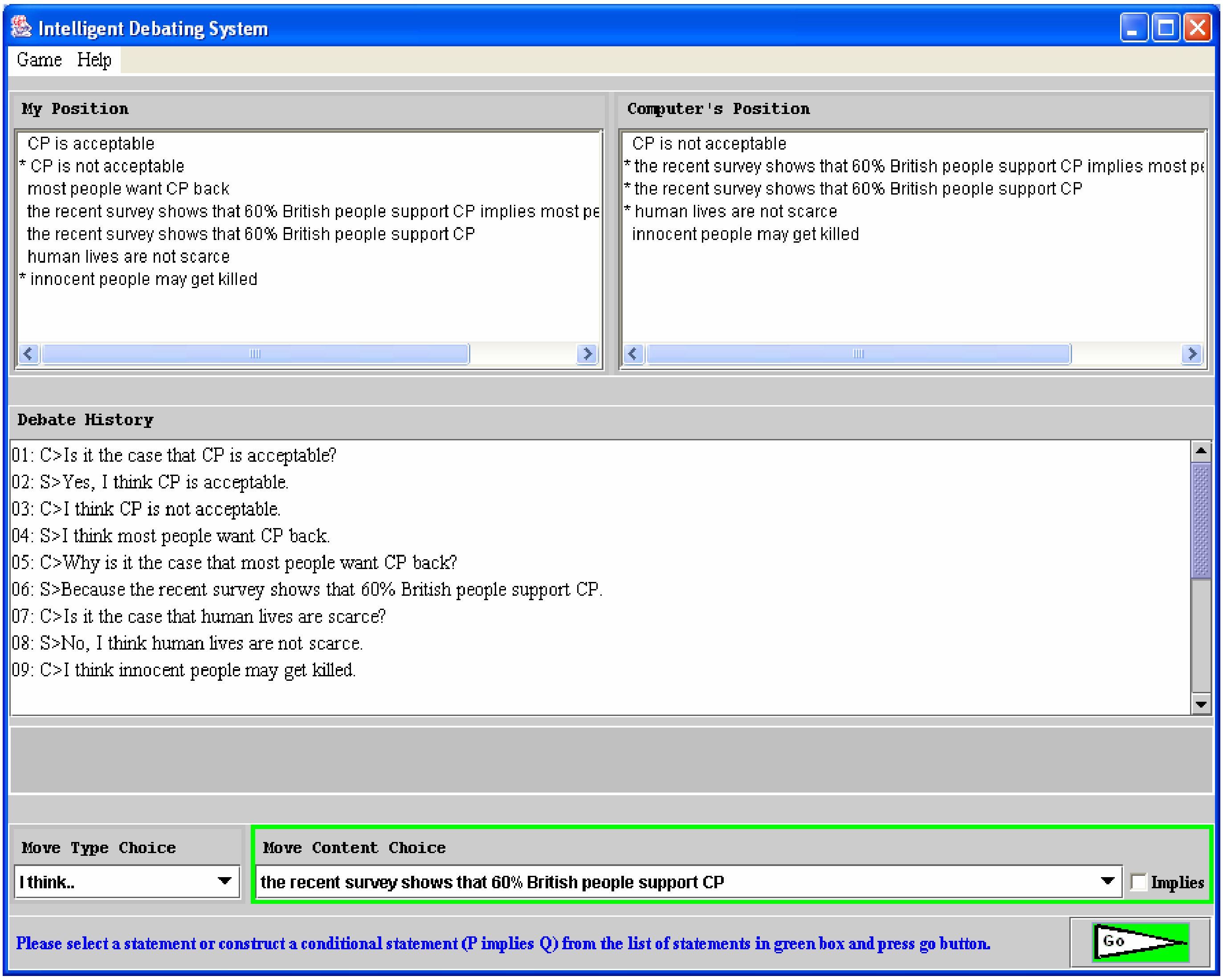

Figure 2 Human-computer Debating System User Interface

The *planning unit* is responsible for generating the computer's dialogue moves. To do so, it takes into account the system knowledge base, the prevailing state of both commitment stores and the dialogue rules. The system planner manages assertion, challenge, withdrawal, resolution and question "strategists", which are designed to deal with different dialogue situations in line with the set of heuristics discussed in section 1. When the system planner receives calls from the dialogue manager, it will check the current dialogue situation and schedule the corresponding strategist to produce a move. Then the system planner will pass the move to the dialogue unit to make the computer's contribution. In addition, there are five components (focus shift manager, build manager, demolish manager, plan generator and plan executor) that are designed to provide special services to the assertion and the withdrawal strategists. The focus shift manager will be called by the assertion or withdrawal strategist to decide whether to change the current focus. The build and demolish manager will be called by the focus shift manager to check whether there are methods available to either build its own position or attack the user's position. The plan generator is responsible for generating a set of propositions and forming a line of questions when required by the assertion strategist, the build manager or the demolish manager. The plan executor is responsible for executing the resulting 'plan'. The assertion and withdrawal strategists will constantly look up whether there is a plan under execution, if there is, then they call the plan executor to carry on its execution.

The *commitment unit* is responsible for updating the user's and the computer's commitment stores. It contains a commitment manager and two commitment stores, one for each party. The commitment manager will update both parties' commitment stores according to the DE commitment rules. Each commitment store is designed to have two lists of statements, those that have been explicitly stated by the owner of the store and those that have been merely implicitly accepted. In the current prototype, a statement that is only implicitly accepted is marked with an asterisk, as shown in figure 2.

The *knowledge base unit* consists of a knowledge base manager and a dedicated system knowledge base. The knowledge base manager provides means of appropriately accessing the knowledge base given requirements from the planning unit.

2.2 Input Facilities

The ambiguity of natural language and the precise input required by the computer renders the use of a general natural language interface currently impossible for the debating system. Moore (1993) argues that a menu driven interface suffices to yield interesting human-computer debate. In addition, there are several precedents for the use of a menu approach in a computational dialogue game (e.g. Hartley and Hintze 1990; Bench-Capon 1998). Although it may be argued that the prescribed menu may not offer the line of argument the user would like to pursue (Moore 1993), this may in practice be alleviated by enlarging the computer's knowledge base and hence providing more available options from which the user may select. In the light of the above arguments, a menu based approach is adopted in the current system.

Under this menu based approach, the user needs to make a double selection, choosing from the available move types and then from the list of prescribed propositions. Moore (1993) suggests that the prevailing set of legally available move types can be identified by the system and only this set be made available to the student. This can be expected to largely prevent the user from breaking the rules. Further, this arrangement may increase the learnability of the game since users are not required to remember the dialogue rules. The approach is therefore adopted by the current prototype.

Once the user has selected a move type, they need to select some propositional content. The system provides a number of means for doing this, depending on the nature of the move type. The details are as follows: (1) the move contents for resolution demand and challenge move types can be selected from the computer's (C's) commitment store; (2) the move contents for a withdrawal can be selected from the user's commitment store; (3) the move contents for assertion and question move types can be selected from the list of propositions (with the aid of the "implies" checkbox shown in figure 2, the user may construct a conditional, e.g. $P \supset Q$). The location of propositions on the screen is highlighted with a green-coloured border. In addition, the message bar at the bottom of the user interface provides dynamic instructions to support user input (see figure 2).

The current system does not allow the user to select a conjunction of statements (e.g. $P \wedge Q$), a disjunction of statements (e.g. $P \vee Q$), or to challenge a conditional. The provision of these is left for further work.

2.3 Knowledge Representation

Several ways of representing knowledge for use in argumentation systems can be found in the literature, e.g. Ravenscroft and Pilkington (2000) use a rhetorical structure theory (RST) and Bench-Capon (1998) and Freeman and Farley (1996) use revised versions of Toulmin's (1958) argument formalism. Moore (1993) argues that the knowledge base should be able to provide answers to questions, support for statements, and rebuttals of statements. To provide such a service for the debating system, the system knowledge base architecture is developed as illustrated in figure 3.

Chaining of argument is allowed by this representation (cf. Bench-Capon 1998). Each proposition might have more than one propositional support. The knowledge base distinguishes hard evidence from opinion; the former is represented by italics in figure 3; the distinction is important in certain aspects of the system's dialogue strategy (cf. Yuan 2004).

The capital punishment domain knowledge is taken from (Moore 1993), an experimental study of the game DC with human participants. The system knowledge base contains a

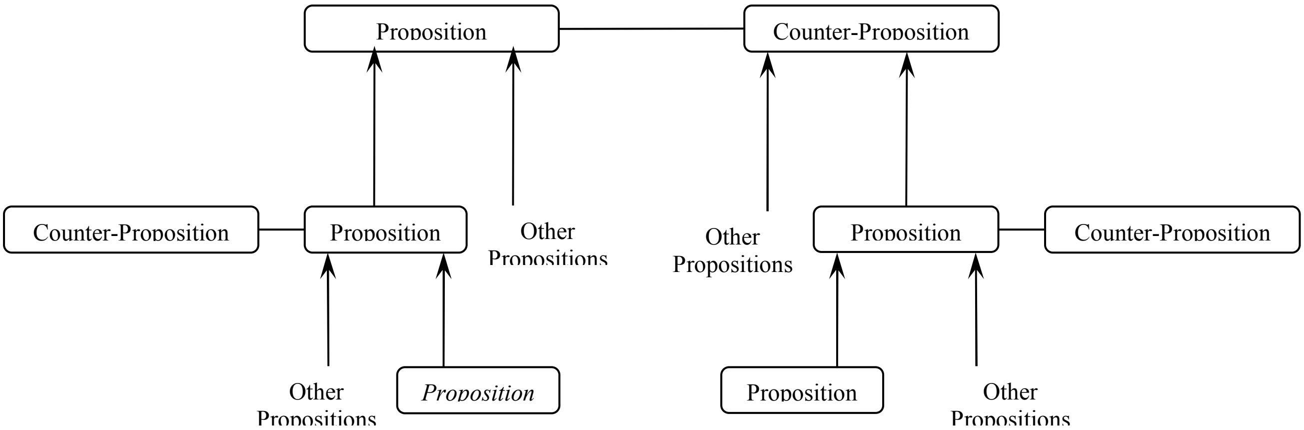

Figure 3 Knowledge Base Architecture

set of propositions and consequence relationships between these propositions; in the current prototype these relationships are based on a structure as shown in figure 3.

3. A USABILITY EVALUATION

A preliminary usability evaluation of the human computer debating prototype has been carried out. Three HCI experts were invited to evaluate the human computer debating system. One expert preferred to evaluate the system cooperatively with the system author, in that the system author noted down the pertinent issues while the evaluator operated on the system (in effect, adopting a cooperative evaluation approach (cf. Dix et al. (2004)). In addition, the expert agreed to take a short interview after the cooperative evaluation session. After the evaluation, the notes of this evaluation were formalised by the system author and emailed to the evaluator to check their accuracy. The two other HCI experts preferred to evaluate the system at their own convenience. The debating system was emailed to these experts. Formal feedback was emailed back to the system author after their evaluations.

Detailed analysis of their comments can be seen in (Yuan 2004). Essentially, the evaluations give positive, albeit preliminary, evidence concerning the usability of the system in general, and of the DE dialogue model and the proposed strategy in particular. This is supported by the evaluators' views on their experiences of operating on the system, such as "definitely easy for students who are familiar with computers", "very straight forward to use it", "no procedures annoyed me while operating on the system", "the system's overall performance is acceptable".

4. CONCLUSION AND FURTHER WORK

We have constructed a human-computer debating prototype using the dialogue model DE and a set of computational strategic heuristics. A usability evaluation has been conducted which furnishes preliminary evidence of the usability of the debating system.

We believe that the work reported makes a valuable contribution to the fields of dialectics and of human-computer dialogue. Concerning the former, we have developed a robust dialogue model and proposed a set of strategies to be utilised with the model. Further, because the computer system we have built can readily be adapted to function with a different dialogue model and/or a different set of strategies, it potentially provides people working in the field of dialectics with a test bed within which they can experiment with new models and new strategies they develop (cf. Maudet and Moore 2001; Amgoud and Maudet 2002).

The work contributes to human-computer dialogue, we argue, in two ways. It indirectly contributes via the contribution to dialectics we have just outlined. Given the usefulness of a dialectical approach to interactive computer systems (cf. Moore 2000, Ravenscroft and Pilkington 2000), any development of dialectics per se potentially has a pay-off in terms of human-computer dialogue. Our work also makes a more direct contribution to human-computer dialogue, in that the debate system is a unique system and therefore makes a contribution to the broadening of the human-computer interaction "bandwidth" in general, and to the development of computer-based education research in particular.

Our immediate future work involves further refinements to the debate system. In particular, we plan to permit freer user input, initially via an option to enter fresh propositional content in addition to selecting from those made available by the system. This will enable us to build up the system knowledge base by adding new claims to it and, more importantly, to experiment with the extent to which the strategic heuristics can cope with such new input.

We also plan further usability studies of the evolving system. The system would ideally be evaluated with a number of different domains of debate, e.g. abortion, politics, terrorism, to test the extent to which the design and knowledge representation are generic. This evaluation might be extended to encompass the use of the system to investigate pedagogic issues (cf. Du Boulay and Luckin 2001), such as the educational value of one to one debate, and how learners make inferences about the knowledge domain.

5. REFERENCES

[1] Amgoud, L. and Maudet, N. (2002). Strategical Considerations for Argumentative Agents. In Proceedings of the Ninth International Workshop on Non-Monotonic Reasoning (*NMR'2002*), Special Session on Argument, Dialogue, and Decision, Toulouse.

[2] Bench-Capon, T. J. M. (1998). Specification and Implementation of Toulmin Dialogue Game. In *Proceedings of JURIX 98*, GNI, Nijmegen, pp.5-20.

[3] Dix, A., Finlay, J., Abowd, G., and Beale, R. (2004). *Human Computer Interaction*, third edition. Prentice Hall.

[14] Du Boulay, B., Luckin, R. (2001). Modelling Human Teaching Tactics and Strategies for Tutorial Systems. *International Journal of Artificial Intelligence in Education,* 12: 235-256.

[5] Freeman, K. and Farley, A. (1996). A Model of Argumentation and Its Application to Legal Reasoning. *Artificial Intelligence and Law,* 4: 163-197.

[6] Hartley, J. R. and Hintze, D. (1990). Dialogue and Learner Modelling. In Cheri, S. A. (eds.) *Student Model Acquisition in a Natural Laboratory (NATLAB),* GEC DELTA Project D-1016 Final Report, Brussels.

[7] Mackenzie, J. (1979). Begging the Question in Non-Cumulative Systems. *Journal of Philosophical Logic,* 8:117-133.

[8] Maudet, N. and Moore, D. (2001). Dialogue games as Dialogue Models for Interacting with, and via, Computers. *Informal Logic,* 21(3) 219-243.

[9] Moore, D. (1993). *Dialogue Game Theory for Intelligent Tutoring Systems.* Unpublished Doctoral Dissertation, Leeds Metropolitan University.

[10] Moore, D. (2000). A Framework for Using Multimedia within Argumentation Systems. *Journal of Educational Multimedia and Hypermedia,* 9(2) 83-98.

[11] Ravenscroft A. and Matheson P. (2002). Developing and Evaluating Dialogue Games for Collaborative E-learning. *Journal of Computer Assisted Learning,* 18: 93-101.

[12] Ravenscroft, A. and Pilkington, R. (2000). Investigate by Design: Dialogue Models to Support Reasoning and Conceptual Change. *International Journal of Artificial Intelligence in Education,* 11: 237-298.

[13] Toulmin, S. (1958). *The Uses of Argument.* Cambridge University Press.

[14] Yuan, T., Moore, D. and Grierson, A. (2003). Computational Agents as a Test-Bed to Study the Dialogue Model "DE", A Development of Mackenzie's "DC". *Journal of Informal Logic,* vol. 23.3.

[15] Yuan, T. (2004). *Human Computer Debate, a Computational Dialectics Approach.* Unpublished Doctoral Dissertation, Leeds Metropolitan University.