A DAI Architecture for Coordinating Multimedia Applications

Keith J. Werkman*
Loral Federal Systems - Owego, Advanced Technology Department
1801 State Route 17c, Mail Drop 0210, Owego, New York 13827
(607) 751-4889 / FAX (607) 751-6025
Email: *keithw@lfs.loral.com*

Abstract

This paper describes a research efforts in exploring various ways distributed multimedia applications can have their presentations coordinated through the use of techniques from Distributed Artificial Intelligence (DAI). DAI problem solving architectures can provide methods for coordinating activities of "multimedia agents" as well as provide mechanisms for conflict resolution among these agents. Conflict occurs when multiple agents request limited hardware resources such as single audio card located in the user's workstation. A novel method of conflict resolution known as Knowledge-Based Negotiation (KBN) is presented which addresses resource allocation problems from an issue-based reasoning approach. Through the use of a distributed problem solving architecture and the KBN negotiation paradigm, the user is assisted in navigating though a multimedia application.

1 Introduction

With the advent of distributed multimedia applications, there is a need for techniques to help users coordinate their activities during multimedia application sessions. As multimedia applications move from the single workstation environment to the distributed, multi-workstation environment, problems of hardware resource allocation and multimedia resource coordination between computer "agents" become more apparent. An example of this occurs during the sequencing of multimedia events which are necessary to give the user a correct impression of the multimedia application's behavior. For instance, an audio card in a workstation may be required by two multimedia agents. The current mode of the application may require a "speech recognition agent" to be accessing the audio card in order to "listen" to the user for speech input. If the user suddenly requests some form of "audio help" by clicking on a help button in the application, the "help agent" might also want to access the same audio card in order to "tell" the user the requested help message. When this happens, the two agents find themselves in conflict over a limited hardware resource. The agents need to be able to successfully resolve their conflicts in order for the multimedia application to continue.

Issues of conflict and coordination of agent activities has been a long time research topic in the field of Distributed Artificial Intelligence (DAI). Usually, DAI systems involve multiple knowledge-based computer agents engaged in distributed problem solving. In order to perform distributed problem solving, intelligent agents need to be able to communicate amongst themselves, coordinate their activities, and negotiate once they find themselves in conflict. Conflict can result from simple limited resource contention to more complex issue-based computations where the agents disagree because of discrepancies between their domains of expertise. An example of this is seen in Werkman's research on conflict among agents during analysis of product designs in collaborative enterprises[1]. Here, a design agent may want a particular issue such as *strength* maintained during the design of the product. Another agent, say a manufacturing agent, might want to substitute an original *strength* issue with a *not-so-strong* part because the substitution leads to a savings in cost or is easier to manufacture for the second agent. The two agents must negotiate the out-

*Part of the following work was performed while the author was employed by the National Science Foundation sponsored Engineering Research Center, Advanced Technology for Large Structural Systems (ATLSS), at Lehigh University.

come of the substitution before they can proceed. It is believed that similar forms of issue-based negotiation will be necessary in distributed multimedia applications in order to resolve not only simple resource allocation conflicts, but also more complex issue-based coordination and presentation conflicts.

An example of an issue-based conflict that might occur between multimedia "navigation" agents while attempting to guide a user through an application is as follows. A "video" agent might be displaying a video when, suddenly, the user speaks "wait." Should the speech recognition agent immediately interrupt the video agent's presentation or wait until the video finishes? This depends on several issues such as the context of the video presentation, the "importance" of each navigation agent, and the required overall appearance or behavior of the application's presentation to the user.

One possible scenario given the user's spoken command is that the video continues to play. Here, the speech recognition agent tells the video agent that the user has spoken something, but the video agent has determined that the video is very important (high perceived user importance) and therefore should continue to play for a while longer. In this case, the video agent tells the speech recognition agent to "hold on" for a while longer. If the user continues to speak "wait" or shouts "stop!" then the speech recognition agent would inform the video agent that possibly the video should be interrupted in order to listen (or appear as though the system is listening) to the user's question. Therefore, the speech recognition agent might send another message to the video agent. This may cause the video agent to relinquish control after determining that a second request had been received, stop the video, and allow the speech recognition agent to become dominate.

Another scenario possibly is that the video agent may reason that its function is more important than the speech agent and therefore refuse to relinquish control. In this case, a third-party "presentation agent" who has been continuously reviewing the situation might providing suggestions to each navigation agent based on knowledge of the application's domain (presentation style). For instance, the application might have noted that video is an important medium which can only be interrupted by a user pressing a button. The "presentation agent," acting like an *arbitrator agent*, maintains basic domain knowledge and knows that all users who shout "stop!" require some form of immediate attention from a speech recognition agent. Therefore, the domain "user-shout-issue" overrules any application presentation issues. In general, the application should provide the necessary *user presentation issues for each media* and the "presentation agent" should determine what is best for the application's presentation to the user given the application's current state. This includes issues such as what is being presented, which navigation agents are present and active, what is the user's model of the presentation, etc.

Werkman presents a novel form of incremental conflict resolution, called Knowledge-Based Negotiation[2], which uses specific aspects of knowledge of the domain to resolve conflicts. Through this negotiation technique, conflicts are resolved by use of shared agent knowledge representations called *sharable agent perspectives.* Agent's perspective are made "sharable" by providing each agent with an indexing scheme of all relevant domain issues. This apriori indexing scheme allows a conflicting agent to relate to another agent's perspectives during counterproposal generation. Thus, each agent can reasonably negotiate without the need for detailed knowledge of the others agents' issues. Pairs of agent relational issues knowledge constructs can then be grouped into a "relational network" which is maintained by a third-party "arbitrator agent". It has been found beneficial to include a third party arbitrator to assist in the negotiation process in order to resolve all issues within a time deadline (necessary for multimedia applications).

In Werkman's scheme, the arbitrator uses the relational network to generate alternative proposal suggestions to the conflicting agents when needed. The arbitrator operates in two phases. In the first, the arbitrator employs simple mediation between the conflicting agents. Here the arbitrator develops suggestions by initially searching a history of past proposals that were generated by each conflicting agent. The proposal history or "negotiation dialog" is searched for relevant issues that exist between the agents. Upon finding a "reasonable" interagent issue relation, the arbitrator presents this shared issue relation to the conflicting agents. The agents can then use the arbitrator's suggestions to identify other possible "viable alternatives" which they might not have considered during their earlier negotiation phase. If the agents still do not agree (mediation fails), the arbitrator enters its binding arbitration phase. Binding arbitration is used to force a "fair" solution on the agents based on their past negotiation dialog. This is necessary due to timing constraints during coordination of multimedia presentations. If events do not occur in a smooth, sequenced order, the user will become confused. Therefore, conflict resolution techniques such as Knowledge-Based Negotiation (KBN) are useful in coordinating the presentation of distributed multimedia applications.

2 The Knowledge-Based Negotiation Model

The knowledge-based negotiation approach used in the KBN architecture requires that agents must be able to communicate the problem, reason from their own and other agent's perspectives, and finally generate a solution. For agents to be able to contemplate the effects of their proposals on others, they need to share a common background of the problem domain. In addition, if agents are to provide some form of explanation of their reasoning of the proposal process, they must be able to communicate issues that the other agents can understand. This is done through the sharing of agent perspectives with other agents. Though sharing common knowledge and agent perspectives, agents can review the dialog of the negotiation and use this knowledge to make better informed proposals at future steps in the negotiation process. The history of the agent dialog also provides a basis for explanations for the user. The negotiation dialog describes the agents' behaviors at each point in time during the negotiation process.

2.1 Agent Communications

Communications within the KBN architecture occurs through a centralized communications medium called a *blackboard* [3, 4] by means of a commonly understood interagent communications language. Because of disparate knowledge between each expert agents, a commonly agreed upon set of communications primitives was developed. This shared vocabulary provides a basis for commonality among the agents during evaluation. As stated in [5], logically valid statements made by one agent in this language should be accepted as valid by other agents. Because of the shared language, an effective form of negotiation between agents can exist. By selecting the right message primitives combined with the right historical dialog (prior proposals and context), agents can communicate abstract levels of intentions and therefore reason about the beliefs of other agents [6]. The interagent language is based on work done in the area of *speech act theory* [7] where speakers perform speech actions like requests, assertions, and suggestions. Cohen and Perrault [8] have suggested that speakers actually use a plan-based approach when making utterances. Thus, the listeners try to infer the intentions of the speaker's speech plan and offer assistance if they can by noting *obstacles* in the speaker's plan. A similar plan-based approach to generating responses to agent queries is used in the KBN architecture. The communications primitives used in the KBN architecture are based on work by Bruce[9] who models *social actions* that occur between people. The social actions used are listed in Table 1.

Table 1: Speech Related Social Actions

Accept	Agent accepts Recipient's cause X.
Ask	Agent doesn't know Recipients cause X.
Command	Agent wants Recipient to cause X.
Convince	Agent convinces Recipient to want X. (Makes Recipient believe he wants X)
Explain	Agent explains lack of outcome X to Recipt.
Inform	Agent informs Recipient of X (simple tell).
Refuse	Agent refuses Recipient's request.
Reply	Agent replies to Recipient's ask.
Request	Agent asks Recipient to want X.

2.2 KB-Negotiation Knowledge Requirements

This section details the types of knowledge required by a knowledge-based negotiation system (KBNS). In a KBNS, three types of knowledge are maintained. The first type of knowledge is *shared knowledge* which is accessible to all agents including the independent arbitrator agent. The shared knowledge includes *domain object knowledge* as well as knowledge about the *history* of the negotiation dialog. Object knowledge includes such things as the names of objects known to all agents in the domain and is predefined and static in the system. The negotiation history includes all agent proposals, rejections, and counterproposals made by the agents and is dynamically generated. The second type of knowledge maintained by the KBN architecture is *unique agent knowledge.*

The third form of knowledge in the KBN architecture includes *knowledge maintained by the arbitrator.* The arbitrator agent maintains both *local* unique agent knowledge related to mediation and arbitration strategies as well as *shared* agent knowledge including

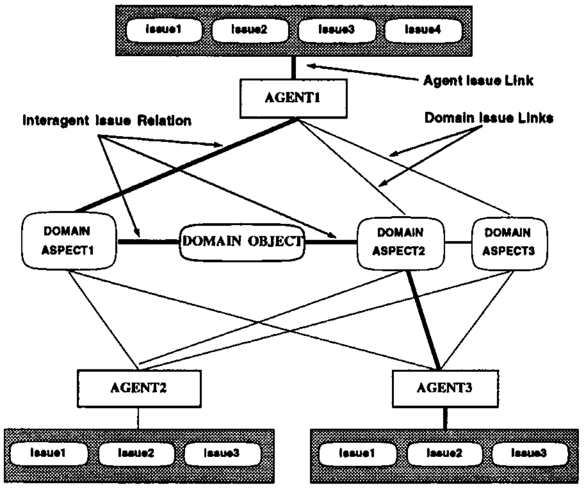

Figure 1: Example Relational Network

object knowledge, agent issue labels and the negotiation dialog. In addition to this shared knowledge, the arbitrator also maintains each agent's *shared perspective* of domain objects. The arbitrator maintains this knowledge in a relational network of **sharable agent perspectives.** An example of a network of interlinked agent perspectives is seen in Figure 1.

The figure contains three *agent nodes* each with its own local *issue nodes* (grey block of issues), linked by *agent issue links.* The two remaining types of nodes in the figure are the *domain aspects* and the *domain object* nodes (boxes with rounded corners). Each *agent node* is linked to a *domain object* by means of a commonly accepted *domain issue link* ¡such as expense¿ which also passes through a *domain aspect* node. This linking of nodes comprises a **sharable agent perspective** in the relational network. When two unique agent perspectives are linked to the same domain object, the results is an **interagent issue relation.** This is seen in the figure as a heavy black line. The arbitrator uses the interagent issue relations to aid in selecting probable alternatives for agent's to consider when they find themselves in a "deadlock" situation and can not develop their own proposals.

The network provides an abstract level description of agent issue interactions that allows the arbitrator to detect immediate conflicts between agents and suggest possible solutions. The proposed network scheme allows for the addition of new agents to the distributed problem-solving model. Initially, each new agent must share its knowledge of relevant issues with the arbitrator so that they can be added to the network and used during negotiation.

2.3 Arbitration in KB-Negotiation

At some point during the agent evaluation and negotiation process, a *proposing agent* might exceed the acceptable limits of the issues of the group. This may require an agent to concede an issue and propose an alternative in order for the negotiation to proceed. It is also possible that an agent may not be able to concede an issue because it would be too costly for that agent. In such cases, the arbitrator agent must be brought in to attempt to mediate a solution between two conflicting agents. Initially, the arbitrator monitors the current status of all agent proposals and reviews each proposal for any immediate problems that they might cause for an agent. If the arbitrator detects a problem that affects a particular agent, it warns the agent and gives control to that agent so that it has a chance to respond to the problem caused by the proposed connection.

In addition to detecting agent problems during proposals, the arbitrator also reviews the history of each agent proposal to determine if a halting condition or a "deadlock" situation has occurred. The arbitrator generates the argument of which issues are relevant from abstract interagent issue relations it obtains from the *KBN Relational network* as well as the history of past proposals and issues. The arbitrator does not contain any knowledge about each agent's unique knowledge. In order for the arbitrator to augment a proposed solution with additional arguments, each agent has to be queried as to the reason and explanations behind the issue relationship under consideration similar to other negotiation systems[10]. A full description of the arbitrator's mediation and arbitration strategies can be found in [11, 12].

In situations where agents still fail to agree after initial negotiation methods, the arbitrator determines the final solution given the input from both agents as to the importance of each agent's issue. This is a form of meta-level control[13] in that the final decision is based on an a priori policy of acceptance specific to the given domain of construction. If the agents' proposals do not converge after six iterations (considered adequate given the evaluation process), the arbitrator stops the evaluation and returns control to the user.

3 Summary

In summary, this paper tried to present the need for Knowledge-Based Negotiation Systems (KBNS) as part of a presentation control methodology for intelligent multimedia multimodal systems. The incremental negotiation scheme called **knowledge-based negotiation** presented here is one potential approach to solving this problem. This scheme allows agents which share a common background to behave both cooperatively and competitively during negotiation, thus providing a form of critical evaluation of the application's presentation style. This is accomplished by use of a shared knowledge representation called **sharable agent perspectives*** A grouping of two of more sharable agent perspectives results in an **interagent issue relation** that relates agents to domain objects by means of **domain aspects**. A network of these relations is maintained by a third party *arbitrator* agent who uses them during its mediation phase of conflict resolution. The arbitrator selects an interagent issue relation based on a review of the *negotiation dialog* for issues that exist between the conflicting agents. The arbitrator suggests the interagent issue relation to the agents in hope that they will consider it as a *viable alternative.* If this fails, the arbitrator enters its arbitration phase of conflict resolution which includes such techniques as setting time limits and searching the negotiation dialog for other proposal alternatives that have similar advantages and disadvantages for each of the conflicting agents.

References

[1] K.J. Werkman. Negotiation in dai as an infrastructure component for collaborative enterprises. In *Proceedings of the IEEE Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE)*, page in press (14p), Concurrent Engineering Research Center, Morgantown, WV, April 1993b. IEEE Computing Society.

[2] Keith James Werkman. *Multiagent Cooperative Problem Solving Through Negotation and Perspective Sharing.* PhD thesis, Lehigh University, 1990.

[3] H. Penny Nii. Blackboard systems, part 2. *AI Magazine*, 7(3):82–106, 1986.

[4] L.S. Baum. *Recent Developments in Blackboard Frameworks*, pages 303–308. Academic Press, Inc., 1250 Sixth Street, San Diego, CA 92101, 1989.

*A comprehensive overview of sharable agent perspectives can be found in [2].

[5] Alan H. Bond. *The Cooperation of Experts in Engineering Design*, pages 462–486. Pitman/Morgan Kaufmann, London, 1989.

[6] Robert R. Tenney and Jr. Sandell, Nils R. Strategies for distributed decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(8):527–538, August 1981.

[7] James F. Allen and C. Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15(3):441–458, 1980.

[8] P. Cohen and R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212, 1987.

[9] Bertram C. Bruce. Belief systems and language understanding. Bbn report no. 2973, ai report no. 21, Bolt Beranek and Newman, Inc., 1975.

[10] Katia P. Sycara. Argumentation: Planning other agent's plans. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 517–523, Detroit, MI, August 1989. IJCAI, Morgan Kaufmann Publishers, Inc.

[11] Keith J. Werkman. *Multiple Agent Cooperative Design Evaluation Using Negotiation*, pages 85–96. Kluwer Academic Publishers, 101 Philip Drive, Nrowell, MA 02061, U.S.A, 1992b.

[12] Keith J. Werkman. *Using Negotiation and Coordination in Multiagent Intelligent Cooperative Information Systems*, pages 251–270. Springer-Verlag, NY, NY, 1992c.

[13] Daniel D. Corkill and Victor R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 748–756. IJCAI, IJCAI, 1983.