Reasoning about Goals to Resolve Conflicts*

Nancy D. Griffeth
Bellcore

Hugo Velthuijsen
PTT Research

May 26, 1998

Abstract

In this paper, we present a mechanism for negotiation to resolve conflicts arising among autonomous agents. These conflicts arise when the activities that the agents need to perform in order to achieve their goals interfere with each other. Two characteristics distinguish our mechanism from other existing mechanisms for negotiation. Our mechanism allows agents 1) to infer other agents' goals, rather than to exchange freely all information pertaining to the agents' goals, and 2) to find alternative activities that achieve the goals of all involved agents, rather than to rely on modification of their own or other agents' goals. These characteristics are obtained by incorporating plan recognition in the negotiation process. The formalism is illustrated using an example where agents representing subscribers to a telecommunications system negotiate over the type of communication session in which their subscribers want to be involved.

1 Introduction

Cooperating systems are bound to run into conflicts over which activities to perform. Such conflicts arise as the systems make incompatible decisions, because they base their decisions on different *information* or because they try to achieve different *goals*. To resolve conflicts, cooperating systems need to interact, exchanging information and possibly changing their own goals or trying to change the goals of other systems. The resulting interactions constitute a *negotiation* process.

Many mechanisms for negotiation among computer systems, or agents, have been proposed in the field of cooperative problem solving (see, e.g., [1, 5, 6, 7, 11, 14, 15, 18, 19]). As these mechanisms were developed for different domains with varying characteristics, each mechanism has distinctive characteristics. Several mechanisms,

*appeared in Proceedings of the International Conference on Intelligent and Cooperative Information Systems, Rotterdam, May 1993

however, are based on a hierarchical representation of goals and alternative ways to achieve these goals [1, 5, 6]. The hierarchy is used for finding a plan that achieves the goals of all involved agents, but that does not involve conflicting activities. Typically, an agent sends other agents information about the goals it tries to achieve and the alternative plans to achieve the goal that are acceptable. This information forms a collection of *proposals* from which the other agents get to pick one that is acceptable to them.

This approach is promising for our own application domain [9, 17] for one particular reason: its ability to find alternative ways to achieve an agent's goals. Unlike many negotiations, where the involved parties may be prepared to change their goals or relax their constraints for possible agreements, we are interested in finding an agreement that achieves — as far as possible — the goals held by the involved agents *without* change.

There is an important limit to this approach, however. Information about which alternatives are acceptable (and which are not) is usually restricted information in our applications: either because it is *strategic* information (used to find an agreement that is most *advantageous* to an agent) or because it is *private* information. The privacy of relevant information is one of the major reasons that a *distributed* approach was chosen, instead of a centralized approach, where all relevant information needs to be made available to at least one not necessarily trusted entity.

We argue that it is not necessary to disclose all the information usually exchanged. The key notion is that goals can be *inferred* from a proposal. This is a task that is performed by *plan recognition*. In fact, Kautz has developed a formalism for plan recognition that makes use of a similar hierarchy as often used in negotiation mechanisms [10]. In this paper, we show how plan recognition can be used to obtain a negotiation mechanism that finds an agreement — if it exists — that achieves all goals held by the involved agents, but limits the amount of information that needs to be disclosed by the agents.

The organization of this paper is as follows. Section 2 briefly describes conflicts in telecommunications systems, the domain for which our mechanism was developed. Section 3 describes our mechanism. A more detailed description of the mechanism can be found in [8]. We conclude this paper in Section 4 and present some directions for future work.

2 Conflicts in Telecommunications Systems

In this section, we will discuss in more detail our application domain, which is the resolution of conflicts in telecommunications systems. We distinguish three ways that conflicts may arise in telecommunications systems. One way is that users may disagree on the particular form of a communication session to be established among them. For example, party A may have an unlisted number, whereas party B may want to see the number of the calling party before accepting any call. When party A calls party B a conflict arises. We call this kind of conflict a *session conflict*, because it involves

disagreement over how the communication *session* should be established. Another example of a session conflict would arise if one party wants to include a third party, while the other does not. More generally, session conflicts involve disagreement over whether to establish communications or what the nature of the communication will be.

A second kind of conflict arises over the use of resources external to the network. For example, if telecommunications user A is at home and B is visiting A's home, calls forwarded from B's home to A's home may conflict in their use of resources with calls to A. We call this kind of conflict a *station conflict*, because it involves equipment at a user *station*.

The third kind of conflict involves the use of scarce network resources. For example, a conference call requires special equipment called a *bridge* that combines the signals from multiple sources into a single signal. If too many conference calls are attempted simultaneously, the network will not be able to provide enough bridges for them. We call this kind of conflict a *resource conflict*.

Some session conflicts and station conflicts occur as a result of what is called *feature interactions* in the telecommunications community. A *feature interaction* arises when one feature of the system interferes with the expected operation of another feature.[1] The feature interaction problem is well-known in the telecommunications community; many telecommunications researchers throughout the world are actively investigating the problem [16, 3, 4, 9].

We have proposed [9, 17] an architecture for telecommunications systems in which each relevant entity (subscriber, resource provider, information provider, etc.) is represented by an agent. The functions of an agent are to produce proposals to set up communication and to evaluate proposals from other agents, possibly generating counter-proposals.

Our negotiation mechanism has been influenced by the demands of the telecommunications domain in two decisive ways: need for restriction of information disclosure and need for achieving each agent's goals. The need for restriction of disclosure of information involves both strategic information (e.g., information directly specifying which proposals will be acceptable to an agent) and sensitive information (e.g., information about policies a subscriber would rather keep private, such as call-screening lists). As an agent is not willing to disclose information pertaining to a conflict freely, other agents that are parties in the conflict have to be able to hypothesize and induce the information they need from the proposals that are received. While a proposal necessarily discloses the information it contains, it does not directly specify what other proposals would be acceptable or unacceptable, or what goals or policies led to this proposal. This observation has led to our incorporation of plan recognition in the

[1] Analogous problems arise when new functionality is added to any software system. Also, there are feature interactions that do not involve conflict between different entities in the telecommunication system.

negotiation process: an agent needs to be able to reason about other agents' goals, in order to be able to modify a proposal in such a way that it achieves its own goals and the goals induced for other agents.

The need for achieving each agent's goals, as much as possible, stems from the fact that in many cases subscribers would rather not participate in a call at all, than to yield some of their constraining policies. Thus, our approach specifically tries to find solutions to conflicts by using what is essentially a *bridging* process, i.e., a process that tries to achieve all agents' goals without change [13].

3 Negotiation Mechanism

We now present a description of our negotiation mechanism. We concentrate on the rationale behind the negotiation mechanism. For a more detailed description of the mechanism we refer to [8]. We first examine a simple negotiation process to illustrate the basic mechanism that is the basis for our approach. Next, we show how the mechanism can be formalized for use in an automated system.

3.1 Analysis of an example negotiation process

We start out by giving an example. Consider a subscriber with an unlisted number, who calls another subscriber with calling-number-delivery (i.e., a subscriber with equipment that allows the subscriber to see the number of the caller). As the subscriber with an unlisted number does not want his or her number to be generally known, this unlisted-number feature seems to be incompatible with the calling-number-delivery feature.

However, we can envision a negotiation session that resolves the apparent conflict. Suppose that the caller with an unlisted number (identified as UN from now on) proposes a call to the subscriber with calling-number-delivery (CND). CND receives this proposal, sees that it does not include the delivery, and returns the previous proposal with calling-number-delivery added. UN receives this proposal, notices the addition, and decides that it is unacceptable. As an alternative UN sends a proposal suggesting delivery of name, instead of number.[2] CND receives this new proposal, decides that it is acceptable, and returns notification of agreement. The following table shows this exchange of proposals.

This example and the fact that an agreement is found hinges on the answer to an important question: why are the proposals that are subsequently exchanged reasonable counter-proposals? Why does UN offer to provide a name upon reception of CND's counter-proposal, instead of, e.g., to pay more, or to include a third party in the call

[2] An unlisted number keeps the link between a name and the corresponding number private. Using only a name for delivery keeps this link undisclosed.

UN	**CND**
connect(UN, CND)	
	connect(UN, CND) & *deliver-number*
connect(UN, CND) & *deliver-name*	
	OK

Table 1: Proposal exchange in our example negotiation.

(each of which may be reasonable counter-proposals in other situations.

The answer lies in the following observation. Typically, a proposal does not represent the ultimate goal of an agent, but merely one way to achieve an agent's goal. There might exist alternative ways to achieve the agent's goal as well. Such alternative ways to receive the same goal provide room for negotiation. To explore these alternatives, it is necessary for an agent that receives an unacceptable proposal to recognize what goal the proposal tries to achieve and to derive from that goal alternative, possibly acceptable, ways to achieve the same goal. If an agent is not informed explicitly what goal another agent tries to achieve, the agent may be able to speculate about the goal, based on the information it does have, i.e., the received proposal.

In the previous example, UN received a proposal to set up a call with number delivery. This proposal in itself is unacceptable. However, in this example, UN assumed that CND's goal was not really to receive UN's number, but rather to receive some identifying information (such as a number). This goal can also be achieved in other ways, e.g., by sending the name instead of the number. As sending the name is acceptable to UN, this alternative is subsequently proposed. Thus, a reasonable counter-proposal to a proposal is one that achieves a same goal as that proposal.

3.2 Automation

In the following, we show how the process of generating counter-proposals described in the previous section can be formalized and automated. The process uses a *goal hierarchy* and a definition of what *acceptable* proposals are. The following sections describe in turn the goal hierarchy, acceptability, and the algorithms that make up our negotiation mechanism.

3.2.1 The goal hierarchy

The relationship between the goal *deliver-number* and the goal *deliver-information* is an *abstraction* relationship.[3] Both goals *deliver-number* and *deliver-name* are special-

[3]An activity can be performed for the sake of the activity itself. In this sense, activities are synonymous with goals.

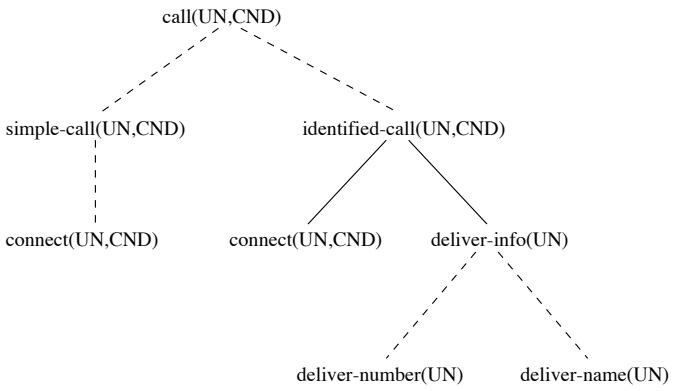

Figure 1: Goal hierarchy for our example negotiation process.

izations of the goal *deliver-information*. Achieving a specialization of a goal implies achieving the goal itself. In our mechanism, we use such abstraction relationships between goals for reasoning about agents' goals and for generating counter-proposals. We also use another type of relationships between goals: *composition*. In the example in the previous section, the goal *connect*(UN, CND) & *deliver-number* is composed of the two goals *connect*(UN, CND) and *deliver-number*. A composite goal is achieved if all its component goals are achieved. Component goals correspond to subgoals in AI planning. Based on these two types of relationships, we can build a hierarchy. Figure 1 shows the hierarchy for our example. The nodes represent goals; the broken lines represent abstraction relationships; the solid lines represent composition relationships.

When one goal is an abstraction of another goal, the other goal is said to be a *specialization* of the first goal. Whenever a goal needs to be achieved, that goal can also be achieved by achieving any of the specializations of that goal or by achieving all of its composite goals. Thus, the abstraction and composition relations define the different ways that a goal can be achieved. We use this notion to define a specification of a goal. A *specification* of a goal is derived by recursively replacing an abstract goal by one of its specializations and a composite goal by the set of its components. In the example of Figure 1, $\{$*connect*$(UN, CND)\}$ is a specification for the goal *call*(UN, CND), as are $\{$*connect*(UN, CND), *deliver-name*$(UN)\}$ and, trivially, $\{$*call*$(UN, CND)\}$ itself. The definition of a specification is such, that whenever all of a specification of a goal is achieved, then that goal itself is achieved.

Kautz uses hierarchies as described above for plan recognition [10]. The task is to recognize what goal a certain agent tries to achieve (and what plan the agent uses) by observing the agent's activities. Intuitively, this process goes as follows. A hierarchy is built containing all the goals the agent can achieve and their interrelationships. Whenever an activity is observed, it can be deduced that the agent is trying to achieve the corresponding goal itself or any of the goals that contain the activity in at least one of their specifications. Several observations may be needed to single out the actual goal that the agent is trying to achieve.

In our negotiation mechanism, we have agents that need to reach agreement over how a call should be set up. Such agreements are stated in terms of operations on a telecommunications system. Operations are similar to activities and can be abstracted and combined into plans that meet the agents' requirements. To reach such agreements agents exchange proposals iteratively until an acceptable proposal is generated (or a timeout occurs). Agents use a hierarchy as described above to generate proposals and to determine whether a proposal is acceptable.[4]

We need to define what a proposal is and what an *acceptable* proposal is. As *proposals* we use specifications (we note that the proposals exchanged in our example are all specifications of the goal $call(UN, CND)$). Whenever a proposal is received, the hierarchy can be used to infer what goals it tries to achieve (for a detailed description of the algorithms see [8]). The following section defines what an acceptable proposal is.

3.2.2 Acceptability

A specification (and thus a proposal) is *acceptable* for an agent if the entity the agent represents (e.g., a subscriber) would agree to it. A specification is *unacceptable* if the entity would not agree to it. Each specification is either acceptable or unacceptable to an agent. However, there are possibly many specifications for each goal, so we ask a subscriber to record the acceptability of goals (a subset of all specifications), not the acceptability of specifications. A goal can be marked either acceptable or unacceptable; or it can be unmarked. When a goal is not marked, we call that goal *indeterminate*. We infer acceptability of a specification from the acceptability of the goals that the specification achieves in the following way.

All specifications of an acceptable goal are acceptable. This means that by marking a goal as acceptable, a subscriber indicates that no matter how a goal is achieved, it will be acceptable. Analogously, no specification of an unacceptable goal is acceptable. A goal with no marking may have acceptable specifications, but also unacceptable specifications. We can use the definition of abstraction to infer acceptability of some goals for which no explicit marking was given:

Inference rule 1: An abstract goal is acceptable if and only if all its specializations are acceptable.

Inference rule 2: Also, an abstract goal is unacceptable if and only if all its specializations are unacceptable.

It can happen that a goal is unacceptable for an agent, while all its component goals are acceptable. For instance, a subscriber may agree to talk to either one of two other

[4]We place certain restrictions on the hierarchy to simplify the algorithms used for generating proposals and for determining acceptability, see [8].

subscribers, but not to both at the same time. Likewise, a goal may be acceptable for an agent, while one of the component goals is unacceptable. For instance, a subscriber may not agree to accept a call from a particular person unless a lawyer is also included in the call. Thus, we cannot use the definition of composition to infer acceptability of goals from the acceptability of other goals. But even if we cannot infer acceptability of goals in such situations, we may be allowed to use heuristic rules to make *assumptions* about acceptability in the absence of explicit assignments for goals. For this, we use the following *heuristic* rules:

Heuristic rule 1: Suppose that a composite goal is neither acceptable nor unacceptable and that all component goals are acceptable, then we assume that the composite goal is acceptable.

This rule states that it is okay, unless explicitly stated otherwise, to assume that something is acceptable, if all activities it involves are acceptable.

Heuristic rule 2: Suppose again that a composite goal is neither acceptable nor unacceptable. If any component goal is unacceptable, then we infer that the composite goal is unacceptable.

By this rule we indicate that something is unacceptable, unless explicitly stated otherwise, if it involves an activity that is unacceptable.

An acceptability marking of a goal hierarchy is made *complete* by repeatedly using the inference rules and heuristic rules presented above, until none of the rules can be applied any more. Completing the marking will, in general, diminish the searching needed for determining applicability of received proposals.

3.2.3 Outline of the negotiation process

We show how a negotiation would proceed for the example presented in Section 3.2.1, using the definitions introduced above:

Specification of policies. The task of the agents in our negotiation mechanism is to reach agreements that achieve short-term goals and meet long-term constraints, or policies. In our example negotiation process, the short-term goal is to set up a call. A policy for UN is that the subscriber's number should not be disclosed. Policies can be specified separately from any particular negotiation process by marking an agent's goal hierarchy with the labels *acceptable* and *unacceptable*. Figure 2 illustrates such markings for UN and CND in our example.

Generating a proposal. The negotiation process is initiated when a goal is identified that needs to be achieved (e.g., in our telecommunications domain by a subscriber indicating that he or she wants to make a call). The goal itself may not

be acceptable, but, assuming that it is also not unacceptable, an acceptable specification of that goal should exist. The hierarchy is searched to find an acceptable specification of the goal. In our example, a specification of $call(UN, CND)$ that is acceptable to UN is {*simple-call(UN,CND)*}. Such a specification is then sent as an initial proposal.[5]

Determining acceptability. When a proposal or counter-proposal is received by an agent, it has to decide whether the (counter-)proposal is acceptable. If it is, the agent can agree to it; if it is not acceptable, an alternative that is acceptable to the receiving agent (and hopefully acceptable to the sending agent as well) needs to be generated. Acceptability is determined by searching the hierarchy. If the proposal is a specification of at least one acceptable goal, the proposal is acceptable. If it is the specification of at least one unacceptable goal, the proposal is clearly unacceptable. If none of the goals achieved by the proposal are acceptable or unacceptable, the proposal itself can still be acceptable. Intuitively, if a proposal can be subdivided into parts each of which achieves an acceptable goal, then the proposal is acceptable. Whenever a proposal is not acceptable, a counter-proposal should be generated. The following examples from Table 1 illustrate acceptability of proposals.

1. The proposal {$connect(UN, CND)$} achieves a goal that is unacceptable to CND and thus the proposal is unacceptable to CND.
2. The returned counter-proposal {connect(UN,CND), deliver-number(UN)} does not achieve a goal that is acceptable to UN, nor a goal that is unacceptable.
3. Finally, the proposal {*connect(UN,CND), deliver-name(UN)*} consists of two goals that are both acceptable to UN, so it is acceptable to UN. It is also a specification of a goal that is acceptable as a whole to CND, so it is acceptable to CND too.

Generating a counter-proposal. When a proposal is not acceptable, we need to find an alternative. We distinguish three cases:

1. The proposal itself may not be acceptable, but sometimes a more detailed specification can be found that is. For instance, the proposal {*connect(UN, CND), deliver-info(UN)*} is not acceptable to UN, but {*connect(UN,CND), deliver-name(UN)*} is.
2. The proposal may achieve a goal that is unacceptable, but possibly also another, not unacceptable goal (e.g., an abstraction of that unacceptable goal) for which an acceptable specification can be found. For instance, the goal *simple-call(UN,CND)* is unacceptable to CND, but its abstraction *call(UN,CND)* does have an acceptable specification.

---

[5] We use a mediator to determine which agents are involved by the proposal and should be included in the negotiation process. Agents send their proposals and counter-proposals to the mediator, which in turn forwards them to the appropriate agents.

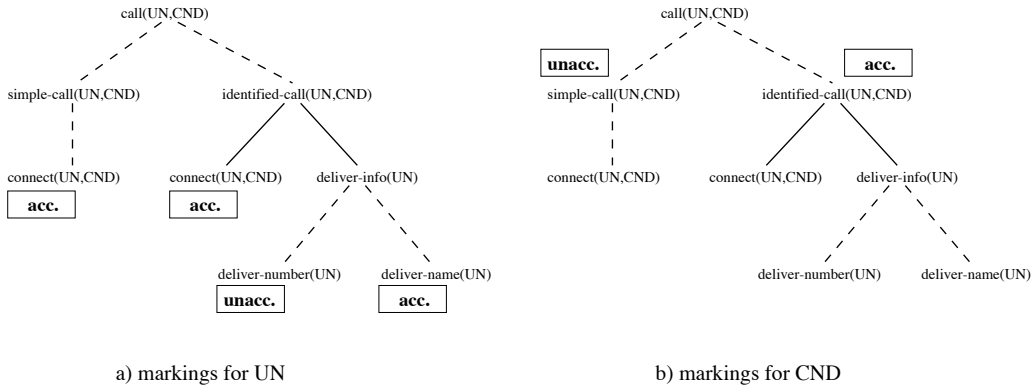

Figure 2: Acceptability markings.

3. If neither of these cases hold, we cannot find a solution without changing the acceptability markings in at least one of the agents. This essentially means that the constraints on the solution must be relaxed by at least one of the involved agents.

The iterative generation of proposals and counter-proposals constitutes a search process through the goal hierarchy to find a specification that achieves the original goal of the agent that initiated the negotiation and is acceptable for all involved parties. The negotiation process terminates when either such a proposal is found, or one of the parties has exhausted all possibilities for generating counter-proposals.

4 Conclusions and Future Work

We have presented a mechanism for negotiation to resolve conflicts among agents. This mechanism uses plan recognition to infer other agents' goals instead of relying on a free exchange of such information. The mechanism searches for agreements that achieve the agents' goals without modification. Central to the mechanism is the understanding of what is a reasonable counter-proposal to a specific received proposal: a reasonable counter-proposal is one that achieves at least one of the same goals of the received proposal.

The mechanism described in this paper is currently being implemented for use in the Touring Machine[TM] research prototype multi-media telecommunications system [2].

There are a number of ways for extending our mechanism to deal with more complex conflict. We present some of the extensions that we are considering for future work.

The current mechanism rates goals and proposals only as acceptable, unacceptable, or

[TM]Touring Machine is a trademark of Bellcore.

indeterminate. Basically, this is equivalent to having a three-valued *utility*. We plan to extend this utility to enable differentiation between proposals that are all acceptable, but not all equally desirable (or all unacceptable, but not all equally dejectable). By using a multi-valued utility it should be possible to rate proposals according to their desirability and to send the proposal first that is most desirable to an agent.

Extending the markings to utilities will also be helpful in another area. Our mechanism now searches for proposals that achieve all agents' goals, and terminates unsuccessfully when no such proposal can be found. We would like to consider next the possibility of changing or dropping some of the goals in order to relax the constraints on possible agreements and to increase the number of solutions. A utility can be used to rate unacceptable goals to determine which one is the least unacceptable and should be changed first.

Our mechanism leaves open such questions as which proposal to send first, or which goal to assume first as another agent's goal. A strategy is meant to make these choices. In future versions of the mechanism, choices will have to be made regarding when to change or drop which goal. These decisions are the task of a *strategy*: a program that uses the elements of our mechanism to implement a negotiation process. A simple strategy would be to take first-in, first-out kind of decisions. In general, the more information a strategy can employ, the more efficient it may become in finding solutions. An example of additional useful information is provided by *critiques* (reasons supplied by an agent why it rejects a certain proposal, [12]). The development of sophisticated strategies will be an ongoing area of research.

The goal hierarchy as currently defined includes only abstraction and composition relationships. These relationships coincide with the mathematical functions AND and exclusive OR (XOR), respectively. We have found certain situations, where the functions OR, MIN, or MAX would have been useful for expressing the relationships between goals. For instance, in the example used throughout Section 3, an alternative to delivering a name or a number might be to deliver both a name *and* a number, meaning that delivering information can be achieved by delivering a name OR a number. This can be expressed using the current relationships, but not without adding a node to the goal hierarchy. In general, restraining oneself to just AND and XOR will increase the number of specializations exponentially in cases where actually combinations of specializations are allowed.

Finally, a question we would like to answer in the near future is: what are the conditions under which negotiation can be successful if the participating agents do not use the same hierarchy?

References

[1] M.R. Adler, A.B. Davis, R. Weihmayer, and R.W. Worrest. Conflict-resolution strategies for nonhierarchical distributed agents. In L. Gasser and M.N. Huhns, editors,

Distributed Artificial Intelligence, volume II, Research Notes in Artificial Intelligence, chapter 7, pages 139–161. Pitman, London, 1989.

[2] M. Arango, P. Bates, G. Gopal, N. Griffeth, G. Herman, T. Hickey, W. Leland, V. Mak, L. Ruston, M. Segal, M. Vecchi, A. Weinrib, and S.-Y. Wuu. Touring Machine: a software infrastructure to support multimedia communications. In *MULTIMEDIA '92, 4th IEEE COMSOC International Workshop on Multimedia Communications*, Monterey, CA, April 1–4 1992.

[3] M. Cain. Managing run-time interactions between call processing features. *IEEE Communications Magazine*, pages 44–50, February 1992.

[4] E. J. Cameron and Y.-J. Lin. A real-time transition model for analyzing behavioral compatibility of telecommunications services. In *SIGSOFT 91 – Software for Critical Systems*, New Orleans, December 1991. ACM SIGSOFT.

[5] R. Clark, C. Grossner, and T. Radhakrishnan. Consensus: a planning protocol for cooperating expert systems. In *Proceedings 11th Workshop on Distributed Artificial Intelligence*, pages 43–58, Glen Arbor, MI, February 1992.

[6] S. Conry, R.A. Meyer, and V.R. Lesser. Multistage negotiation in distributed planning. In A.H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 367–384. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[7] R. Davis and R.G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.

[8] N.D. Griffeth and H. Velthuijsen. Conflict resolution in telecommunications systems: a formalism for proposal generation. TM-ARH-021-725, Bellcore, Morristown, NJ, August 1992.

[9] N.D. Griffeth and H. Velthuijsen. The negotiating agent model for rapid feature development. In *Proceedings of the Eighth International Conference on Software Engineering for Telecommunications Systems and Services*, Florence, Italy, March/April 1992.

[10] H.A. Kautz. A formal theory of plan recognition and its implementation. In J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D. Tenenberg, editors, *Reasoning about Plans*, chapter 2, pages 70–125. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1991.

[11] S. Kraus, E. Ephrati, and D. Lehmann. Negotiation in a non-cooperative environment. *Journal of Experimental and Theoretical Artificial Intelligence*, 1:255–281, 1991.

[12] S.E. Lander and V.R. Lesser. Negotiated search: cooperative search among heterogeneous expert agents. In *Working Notes AAAI Workshop on Cooperation among Heterogeneous Intelligent Systems*, pages 74–83, San Jose, CA, July 1992.

[13] D.G. Pruitt. *Negotiation Behavior*. Academic Press, New York, NY, 1981.

[14] A. Sathi and M.S. Fox. Constraint-directed negotiation of resource reallocation. In L. Gasser and M.N. Huhns, editors, *Distributed Artificial Intelligence, volume II*, Research Notes in Artificial Intelligence, chapter 8, pages 163–193. Pitman, London, 1989.

[15] K.P. Sycara. Argumentation: planning other agents' plans. In *Proceedings IJCAI-89*, pages 517–523, Detroit, MI, 1989.

[16] *Proceedings of the 3rd Telecommunications Information Networking Architecture Workshop (TINA-92)*, Narita, Japan, January 1992.

[17] H. Velthuijsen and N.D. Griffeth. Negotiation in telecommunications systems. In *Working Notes AAAI Workshop on Cooperation among Heterogeneous Intelligent Agents*, pages 138–147, San Jose, CA, July 1992.

[18] F. von Martial. Coordination by negotiation based on a connection of dialogue states with actions. In *Proceedings 11th Workshop on Distributed Artificial Intelligence*, pages 227–246, Glen Arbor, MI, February 1992.

[19] G. Zlotkin and J.S. Rosenschein. Negotiation and conflict resolution in non-cooperative domains. In *Proceedings AAAI-90*, pages 100–105, Boston, MA, 1990.