A Negotiation Model for CSCW

Manuel Munier[1,2], Karim Baïna[1], and Khalid Benali[1]

[1] LORIA - INRIA - UMR 7503
BP 239, F-54506 Vandœuvre-lès-Nancy Cedex, France
{baina,benali}@loria.fr

[2] LIA - UPRES EA N°21.63
BP 1155, F-64013 Pau Cedex, France
munier@marsan.univ-pau.fr
(since september 1999)

Abstract The aim of this paper is to present our model for a generic and flexible negotiation service[1] and its implementation in Computer Supported Collaborative Work (CSCW) environments. Within a cooperative work environment, users naturally need negotiation support mechanisms to study possible alternatives in group decision making. The objective of our work is to build a negotiation model independent of any particular application field. Contrary to studied models, our model focuses on formalizing the negotiation from three points of view: exchanged information between the agents to negotiate (the language), the way this information is exchanged (the protocol), and the internal behavior of an agent (the tactics). In addition to a separation of the problems involved in each one of these three facets of the negotiation, this approach allows also a greater flexibility than traditional systems dedicated to one kind of problems. We chose to use a transactional approach based on speech acts to develop our axiom based negotiation model which has been implemented as a negotiation service in our CSCW environment *DisCOO* [1, 2]. After the tackled problem presentation, we will expose the state of the art, then the proposed negotiation formal model, and, finally, the implementation.

1 Introduction

Nowadays, the explosion of distributed applications for telecomuting, cooperative work, video-conference, electronic commerce,..., is giving an increasing importance to computer supported negotiation. Within a cooperative work environment, group decision making is among executive costliest tasks and users need some negotiation support to re-frame conflictual purposes, identify and structure different problems, propose, attack and defend solutions, and, finally, reach and confirm agreements [3, 4].

By integrating a negotiation support service within a cooperative work environment, we aim to increase cooperation among users by allowing each of them

1. This material is based upon work supported by the Xerox Research Center Europe in Grenoble and LORIA-INRIA.

to express his/her opinion inside the group decision. Such a negotiation service could offer, for instance, a comfortable framework to assist users in deciding which legacy tools to use for the elaboration of shared resources (ex: CAD applications, text editors, compiler and libraries versions), which resources to share with which partner, which cooperation schemas to share a resource, the final version for a shared resource (when modified by several users), how to merge two or more resources to obtain a common resource,...

Negotiation has been discussed by several research fields like psycho-sociology, game theory, economics, artificial intelligence. The objective of our work is to build a negotiation model independent of any particular application field: we do not want to produce yet another proprietary system, but to define a framework which can be configured to support various forms of negotiation.

This paper is organized as follows. After the tackled problem presentation, section 2 exposes the state of the art, section 3 describes the proposed negotiation formal model, finally, section 4 presents the implementation.

2 State of the art

There exists no universal negotiation model we can apply to every negotiation problem. The main reason is the multitude of parameters depending on culture of negotiators [5], language and vocabulary they use to negotiate, field of negotiation, and media of communication. The complexity of a negotiation process depends on negotiator number, negotiator coalition number, conflict nature, force relations between negotiators, perception of the conflict to resolve, purposes to reach,... Aiming to formalize a generic and flexible negotiation model and achieve a negotiation support service for a CSCW environment, we will discuss negotiation from the point of view of Distributed Artificial Intelligence, Speech Act theory and Group Decision Support Systems.

Distributed Artificial Intelligence (DAI) is the discipline that aims to design a system as a community of agents working together to resolve a problem. An agent is a particular computational unit which has a state, develops an expertise, and is able to react to external events. Negotiation techniques (based on theoretical frameworks like BDI[2] model [6, 7]) can be useful either in conflict resolution (to limit their effects by establishing compromises or by going beyond the nature of conflicts) or in service exchange situations (a service request may concern a failure diagnosis, a purchase of market parts, an allocation of a bandwidth for video-conference transmission,...). The *Contract Net Protocol* [8] dealing with the distribution of tasks between agents illustrates an example of service oriented negotiation. The *Commitment-Based Communication Model* [9] upgrades the Contract Net Protocol by allowing more interactivity and transparency between agents (possibility of compromising, arguing, counter-proposing,...). The negotiation process continues until one of the servers accepts a proposition.

Other works related to the communication between agents have proceeded from the point of view of negotiation language. While the use of natural lan-

2. BDI: Believes, Desires, and Intentions

guage is unlimited, the Speech Act theory [10–12] asserts that it is possible to cluster terms in a finite number of verbal actions groups: *speech acts*. This theory introduces the notion of communication protocol to represent different social agents behaviors. As detailed in [13], the Speech Act theory has numerous applications in distributed systems, natural language processing, electronic data exchange protocols,... Many DAI approaches use this theory to formalize inter-agent communication protocols, especially to model the *illocutionary* aspect of speech acts (expected effect of messages). [13] proposes a classification in seven categories: *assertive*, *expressive*, *declarative*, *promissive*, *permissive*, *directive* and *prohibitive* acts. [14, 15] use speech-acts to model process centered cooperation as a four-phase loop workflow (request, negotiation, performance, satisfaction). In Speech Act theory, negotiation processes were modeled either by state/transition diagrams where transitions are labeled by speech acts or by sets of ECA rules [3] in which an event corresponds to the reception of a speech act and an action describes the reaction to this act (sending of an other speech act) (cf. Information Lens [16], SANP [4] [17]).

Finally, Negotiation Support Systems (NSS), which are a sub-class of Group Decision Support Systems [5] [18], combine communication, computing and decision support techniques to assist people in their negotiation tasks. Within conflictual situations, they can be used as a shared and common language for mutual understanding. They traditionally support the process of negotiation rather than model the decision-making aspects of the problem [19]. NSS can provide either an automated or a semi-automated assistance to express the problem in a transparent and structured manner. Services ensured by GDSS/NSS environments can be either synchronous or asynchronous. On the first hand, synchronous services develop rendez-vous mechanisms, requiring the "co-presence" of different actors before the negotiation can begin (i.e. video-conference, electronic meeting systems,...). On the other hand, asynchronous services, offer a more flexible negotiation framework in which actors can interact and work as they do usually (eg: electronic mail, Computer-Mediated Communication, web-based negotiation [5]).

While they propose a well defined framework to model negotiation process, the negotiation models and techniques we have presented here suffer some drawbacks as far as our objective is concerned. For instance, many of DAI models (including game theory) are based on heuristics and utility functions allowing agents to estimate profits of proposed alternatives. Since, this approach ensures the termination of a negotiation (each proposition is *better* than the previous, negotiation by successive refinements,...), it assumes the existence of an order relation (at least partial) on the set of possible propositions. This assumption may be too strong in many application fields. Moreover, the design of most of negotiation services do not distinguish the purpose of negotiation from the

3. ECA: event/condition/action
4. SANP: Speech-Act-based Negotiation Protocol
5. GDSS: Group Decision Support Systems is a discipline gathering technologies and methodologies helping groups of people to express and resolve unstructured problems

negotiation protocol (negotiation mechanisms are hard-coded among the other features of the application). Thus these negotiation services are very specific to the application field. While GDSS and NSS research provide rich notions to analyze and model negotiation process [20–22], they cannot be easily integrated within a CSCW environment. Either they give an informal expression framework which cannot be semantically usable inside a CSCW environment (electronic mail, video-conference, Computer-Mediated Communication, SANP), or they are adapted to the resolution of very specific problems (Information Lens [16], INSPIRE [5]).

Contrary to studied models, our approach distinguishes three different parts within negotiation mechanisms: data structures to be exchanged between agents, communication protocol between agents and their behaviors face to other negotiator acts. Such an approach provides us with a considerable flexibility since it will be possible to adapt negotiation mechanisms to take into account the negotiation context (topic of negotiation, agents roles,...). In such a way, several negotiation modes can be supported within a single CSCW environment. Finally, these interactions between agents can occur either in a synchronous or an asynchronous manner.

3 Axiom based negotiation model

We place ourselves within the framework of a distributed system composed of autonomous agents. An agent can be active (it reacts automatically or semi-automatically when it receives messages from the other agents) or passive (it acts under the control of a user who executes methods on the agent). Thus, the negotiation can be seen in the following way: *"An agent requires something from a second agent, and these two agents dialogue to find the best possible answer"*. The objective of our work is to build a negotiation model which is independent of any particular application field. It means defining **generic mechanisms for negotiation** that one can "instantiate" to negotiate the access rights to a particular service, or to define the cooperation rules to be respected when two agents share a resource, or to any situation requiring a negotiation process.

3.1 Overview

From the negotiation point of view, the two implied agents play different roles: *client* and *server*. The client is the agent having a problem to solve. For this purpose, it carries out a *request* to one of its partners (called server) to find a *solution* for its problem. It is the client which initiates the negotiation with the server. The request is the description of the problem that the client wants to solve. According to the application field, the problem may concern the use of a service, the sharing of a resource, the selection of a resource among those available,... The solution describes the alternative proposed by the server to satisfy the client's request. Depending on the application field, the solution may designate various concepts: access rights for the use of a service, cooperation rules to control the sharing of a resource, designation of a resource according to negotiated selection criterion,... A solution corresponds to a possible agreement between both agents. As we can note, the terms client and server only indicate roles played by the various agents during a given negotiation, and more particularly at the time of their connection. We are in fact in a peer-to-peer organization, i.e. an agent will be at the same time client and server according to whether it sends a request to an agent or it answers the request of one of its partners (i.e. it proposes a solution).

We can split the process of bilateral negotiation into three stages. First of all, the client establishes a connection with the server and announces its request. It is the **solicit** phase. Then the **negotiation** phase itself is held: the client and the server exchange messages to build the set of solutions acceptable by both the agents. Finally, the client chooses one solution among those resulting from the negotiation phase and informs the server. It is the **evaluate** phase. Another goal of this phase is to ensure that the client chooses at most one solution among those negotiated. The server can then performs actions corresponding to the selected solution (allocation of requested resource, allowing accesses to reach a service or to share data,...).

Our work focuses on the negotiation phase formalization. For that purpose, we chose to tackle the negotiation from three points of view for: the information exchanged between the agents to negotiate (the **language**), the way this information is exchanged (the **protocol**), and the "internal" behavior of an agent (the **tactics**). In other words, we distinguish the language in which are represented the decisions taken by the agents, the protocol allowing the agents to communicate their decisions, and the way in which an agent makes its decisions (own objectives, constraints to be checked, semi-automatic reactions to the decisions of the partner,...). More than a separation of the problems involved in each one of these three facets of the negotiation, this approach allows also a greater flexibility than traditional systems dedicated to one kind of problems (systems based on hard-coded negotiation protocols). Following our approach, one may combine various languages, protocols and tactics within a same community of agents, or even within a same agent if it simultaneously carries out several negotiations with several partners.

3.2 Using speech-acts to negotiate

As we already explained, we do not want to produce yet another proprietary system. On the contrary, our aim is to define a framework which can be parameterized to support various forms of negotiation. To achieve this goal, we chose to use a transactional approach based on speech acts: a negotiation is seen as being a transaction having a beginning (*solicit*), an end (*confirm* or *kill*), and having to respect certain criteria. We journalize (or log) some of events (in particular speech acts events via the invocation of $assert[speech\_act]$ operations) on the level of each agent (concept of local history), then we define properties which have to be checked on these histories (concept of correctness criterion). Some of these properties are dedicated to the coordination of the events between two agents (negotiation protocol), whereas some others relate to the control of the decisions taken by an agent (tactics implemented by this agent).

One should note that within this framework each agent has its own local history in which it only journalizes the events relating to itself, namely its own operations like its interactions with the other agents of the system. Once the properties are defined, each agent will be able to locally evaluate them according to its own history. Such an axiom based approach where axioms are locally checked by each agent according to a partial view of the system has several advantages:

- **Correctness**: The control of the negotiation process is based on the definition of properties on allowed sequences of decisions taken by the agents (correctness criterion) and not on predefined scenarios as with states/transitions diagrams. Axioms are viewed as pre-conditions on events invoked by agents.
- **Genericity**: The proposed formalism for defining these negotiation mechanisms is independent of any particular application field.
- **Distribution**: Each agent itself controls the "correct" executions of the negotiation process according to the information journalized (logged) in its local history (peer-to-peer organization between autonomous agents).
- **Scalability**: One can add new axioms without any risk of combinatory explosion as with the integration of numerous states/transitions diagrams.
- **Heterogeneity**: Within a single system, all the agents are not forced to negotiate according to a single model. Moreover, one agent can simultaneously lead in parallel several negotiations according to different models.

The originality of our approach is that controls made by the protocol and the tactics on the decision sequences are not founded on predefined negotiation scenarios (cf. state/transitions diagrams) but on the definition and the checking of invariants characterizing the decision sequences regarded as "correct". Using again the transactional systems vocabulary, one may say that we use negotiation correctness criteria. In order to illustrate our model, we will give two examples: the first one represents a negotiation protocol, the second one a negotiation tactic. Although these two examples were formally specified, we will only give a broad view of them, the objective of this article being the presentation of our approach rather than a formal and exhaustive definition of the operations and axioms of the model itself.

A Negotiation protocol sample The objective of a negotiation protocol is to control the interleaving of the *assert* operations invoked by both agents. That can be seen as a problem of access concurrency on the communication channel or as a problem of speech rights management: the rules controlling decision sequences, the possible decisions according to the context (current history), the actions to be undertaken according to the decisions taken by the partner,... do not relate to the negotiation protocol but to the negotiation tactics.

One example of negotiation protocol between two agents is the **turn-taking**. This one ensures that an agent will be able to make a decision, i.e. to invoke an $assert[speech\_act]$ operation, if and only if it has previously obtained the right to speak ($getTurn$ event). Thus, in addition to the various $assert$ operations invoked by it or by its negotiating partner, an agent will also journalize other events such as $getTurn$ in his local history. The main axiom of the turn-taking protocol can then be schematized by the following invariant on the local history of each of both agents: *"an agent A can make a decision ($assert_A[speech\_act]$) if and only if it acquired the right to speak ($getTurn_A$ appears in its local history) and if no other agent $B \neq A$ acquired this right since then (i.e. no $getTurn_B$ event was journalized since the last $getTurn_A$ event)"*. If the local history of agent A is denoted by H_A and if the relation of precedence between the events is denoted by $\rightarrow$, this axiom can be written:

$$(assert_A[speech\_act] \in H_A) \Rightarrow \left\{ \begin{array}{l} \quad getTurn_A \in H_A \\ \wedge \quad \forall\, B \neq A \quad \nexists\, getTurn_B \in H_A \\ \qquad\qquad\qquad \text{such that} \quad LastOcc(getTurn_A) \rightarrow getTurn_B \end{array} \right.$$

A Tactical sample The concept of negotiation is not limited to agent communication aspects (language and protocol) nor to a single problem solving. It is also necessary to formalize the "internal" behavior of an agent, either from the point of view of the strategy adopted to negotiate with a partner (reactions, concessions, decision changes, ...) or from the point of view of the policy implemented to coordinate several negotiations. The objective of a negotiation tactical is to define the way an agent will make its decisions and/or will choose the solutions ($assert$, $reserve$, $confirm$, $kill$, ... operations) in terms of properties applied to its local history. Such a tactical enables us to describe:

- The **contextual behavior** of the agent, i.e. the decisions which can be taken relatively to the decisions which were previously taken by the various actors of the negotiation (journalized in its local history).
- The **reactive behavior** of the agent, i.e. the decisions which must be taken in reaction to the decisions taken by the partners.
- The **coordination** of the various negotiations carried out by an agent: when an agent is negotiating with several partners, these negotiations are not independent and the negotiated solutions are constrained one by the others (see an example in [23]).
- The **objective** of the agent, i.e. the constraints on its decisions: for example, if we use a utility/interest/cost/... function, a rule can be to always improve a certain coefficient in comparison to the preceding decisions aiming to ensure the convergence of the negotiation (cf. Operational Research, IAD).

A negotiation tactical can thus be seen as a set of behavior rules controlling the sequence of the operations invoked by an agent. To express these rules as axioms, the only available information is the events journalized in the local history of this agent.

One example of negotiation tactical is the **atomic negotiation**. A client agent takes part simultaneously in several negotiations with the constraint that either all the negotiations lead to the implementation of a solution, or no action is undertaken. This behavior is formalized by the both following axioms in which

RS (*RequestSet*) denotes the set of atomic negotiation requests. The first one indicates that an atomic negotiation solicited by an agent A is in fact made up of several elementary negotiations with agents $B_1, \ldots, B_n$.

$$(solicit\_atomic_A[RS] \in H_A) \;\Rightarrow\; \forall\, req_{B_i} \in RS \quad (solicit_A[req_{B_i}] \in H_A)$$

The second axiom affirms that if the agent A has confirmed a solution for one of the requests (with a agent B_i), then it must have also confirmed a solution for each other request (with agents B_j).

$$(solicit\_atomic_A[RS] \in H_A) \;\Rightarrow\; \forall\, req_{B_i} \in RS$$
$$(confirm_A[req_{B_i}] \in H_A) \;\Rightarrow\; (\forall\, req'_{B_j} \in RS \quad (confirm_A[req'_{B_j}] \in H_A))$$

This tactical corresponds to some extent to the two phases locking mechanism used in traditional transactional systems: a transaction requiring the use of several resources must first of all acquire a lock on each resource (first phase) before being able to use these resources (second phase). If, during the first phase, one of the resources cannot be locked, then all the already acquired locks are freed and the transaction is postponed and started again later on.

3.3 Expressiveness of our model

We think however that the negotiation between agents is not limited to the definition of a language, a protocol and tactics. In order to be able to support the various forms of negotiation, other concepts must be associated with these basic mechanisms. One example is the concept of **alternative branches** which represents the fact that both negotiator agents explore several parallel ways to find (different) solutions for the same problem. Another example is the concept of **delegation** which allows a server agent to bypass the control to a second server agent to continue the negotiation with a client agent. From this point of view, the fact that we check axioms on the local histories of each agent allows a greater flexibility than the classical approaches based on predefined explicit workflows. To integrate such new functionalities into our model we only need to define the new necessary events as well as the axioms controlling their use. If needed, it will be necessary to analyze the interactions of these new axioms with the existing ones in order to be sure that the coherence of the set of rules is preserved (for instance, we have to avoid any contradiction between axioms).

Alternative branches During negotiation, it may happen that it is no more possible to define any general solution to the whole problem. It is then necessary to make assumptions before being able to continue. These assumptions permit then to explore various alternatives, i.e. to continue the negotiation in parallel towards several different directions. Let us take an example in which an agent wants to access to the source code of a program developed by another agent. During the negotiation, the server agent proposes two choices to the client agent: the currently available public version, as well as a new version still under development, but access of which is subjected to the signature of Non-Disclosure Agreement. Before choosing, the client wishes to study both alternatives in parallel in order to evaluate benefits and drawbacks for each choice. If it decides to use the public version, it knows that it will have to migrate towards the new version as soon as available. On the contrary, if it chooses the new version under development, isn't the signature of the NDA likely to disturb its relations with other partners? Concerning the creation of a new branch of negotiation, we chose to represent this event by a particular decision, i.e. by a specific speech act which will appear in an *assert* operation. In this manner, it will be automatically journalized in the local history of the two negotiating agents.

Even if both agents can negotiate by simultaneously exploring several alternatives, one and only one solution can be finally chosen by the client. The controls carried out during the evaluation phase of a solution prohibit selection of multiple solutions on a given server for a same request, even if these solutions arise from different branches of negotiation.

Delegation Another classical concept in negotiation is the delegation. For an unexpected reason, the server cannot (or does not want any more) continue the negotiation with the client and decides to give the control to another server. This may concern a particular branch or all the branches of the negotiation. The delegation holds in two phases: the initial server must first of all stop its negotiation (or negotiation branch) in progress, then the client negotiates with the new server. However, it is not simply a *kill* operation followed by a *solicit* operation starting a new negotiation from zero. We consider this change of interlocutor as a "normal" decision in the negotiation process (*delegate* speech act).

3.4 Synthesis

The presented negotiation model allows to coordinate negotiations between agents, this coordination being based on invariants to be respected by negotiation histories. This coordination is definitively not constrained by a completely predefined scenario. This approach provides more freedom to agents: the model does not impose to take a decision, or to forbid a decision. It simply controls that the freely taken decisions are coherent. Nevertheless, helping agents in their decision making process is possible through the use of a negotiation tactical. Such a negotiation tactical may ensure convergence of negotiation process, or may forbid to cancel previously taken decisions,...

The following section presents implantation of this negotiation model on top of *DisCOO*, our CSCW environment. *DisCOO* kernel's aim is to coordinate exchanges of documents between several users. This coordination ensures that these exchanges respect certain cooperation schemas. Negotiation mechanisms have been added to assist users to resolve conflicts when sharing documents, or to simply assist them in the choice of the cooperation schema they agree to respect for sharing a specific document.

4 Implementation

We started this work on negotiation after the development of *DisCOO* [1, 24], our CSCW environment that allows geographically distributed agents (activities, driven by human operators) to cooperate by exchanging resources (mainly documents). Each agent having its own copy of the shared resources, the cooperation between agents is carried out through imports and exports of resources. When two agents connect together to share a resource, they must initially sign a contract indicating the cooperation mode that they chose ("cooperative-write"[6], "client/server"[7], and "writer/reviewer"[8]). Thus, it is indispensable to allow the agents to negotiate this cooperation mode according to their respective constraints. In addition, it can be useful to renegotiate a cooperation mode in runtime to solve a conflict between two modes or to soften the relations between two agents. Finally, these same negotiation mechanisms can be also used to assist the agents when they merge two versions of a same resource, typically in the final phase of a cooperative-write relation.

This implementation within *DisCOO* enabled us to validate the basic concepts of our model, namely the negotiation language and the negotiation protocol, about an example: the negotiation of the cooperation schema to be used for the sharing of a document. While the decision making lies with human operators supported by artificial agents, various informations are available in the local histories of each agent to provide more assistance, control and/or automation of the decision-making process. This will constitute the second stage of the implementation of our model within *DisCOO* via the definition and the implementation of negotiation tactics.

5 Conclusion

In this paper, we presented a brief description of the state of the art in negotiation systems. An analysis of current negotiation systems or approaches lead us to conclude that current systems/approaches are inflexible, lack any genericity across application field, and do not adequately answer to negotiation needs in Computer Supported Collaborative Work. This misfit between current solutions and negotiation needs in CSCW leads us to propose an original solution for a generic negotiation model for CSCW. This model focuses on formalizing the negotiation from three point of views: the language (negotiation information), the protocol (the way negotiation information is exchanged) and the tactics (the way a negotiator takes its decisions). From a modeling point of view, our approach provides an interesting alternative to current negotiation approaches and allows

6. Cooperative Write: two agents modify concurrently the same resource; both must reach a consensus on the final value of this resource.
7. Client/Server: an agent modifies a resource and another agent recovers, progressively, the different produced versions; this cooperation mode ensures that the customer will take well into account (at least) the last version produced by the server.
8. Writer/Reviewer: an agent A produces a document doc_1 and shared it with an agent B (B is client in a client/server relation with A); agent B uses doc_1 to produce a document doc_2 that it shares with agent A (B is server in a client/server relation with A); this cooperation mode guarantees that doc_1 and doc_2 will be coherent at the end of the process.

a greater flexibility than traditional systems. We formalized our axiom based approach upon a speech-act transactional approach. In our model, each negotiation information is represented as a speech act and a negotiation is seen as being a transaction which must respect certain criteria. Each negotiator agent journalizes (or log) the events relating to its own operations and to its interactions with the other agents. We defined and presented properties that have to be checked on these histories, some of the former being dedicated to control the negotiation protocol, whereas others are dedicated to control the taken decisions. The fact that we use axioms to model negotiation properties allows us to enrich our negotiation model with new axioms aiming to tackle classical negotiation need such alternatives or delegation. This enrichment can be pursued as long as the coherence of the set of rules is preserved and may be an interesting extension of our work. We validated our approach through the implementation on top of our CSCW environment (*DisCOO*) of an example using our model's basic concepts, namely the negotiation language and the negotiation protocol. Being mainly interested by human actors, we neglected the negotiation tactics assuming that the human actors will tackle this aspect without any help. Nevertheless, tackling this part of our negotiation model and implementing it will allow us, in the near future, to program reactive agents which can assist negotiators as long a s their tactics have been formally described. Another research direction in which we intend to use our results in negotiation modeling is dynamic federation of cooperative workflows. To allow cooperation and henceforth coordination of different workflows, various negotiation must hold between activities of different workflows (eg: rendezvous, contracts, results needed, exchanged services).

References

1. M. Munier, K. Benali, and C. Godart. A transactional Approach for Cross-Organizational Cooperation. In *GlobeComm99 (Enterprise Applications and Services Symposium)*, Rio de Janeiro, Brazil, december 1999.
2. M. Munier and C. Godart. Cooperation services for widely distributed applications. In *Tenth International Conference on Software Engineering and Knowledge Engineering (SEKE'98)*, San Francisco Bay, USA, 1998.
3. M. Stefik, G. Foster, D. G. Borrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1), January 1987.
4. L. L. Putnam and M. S. Poole. Conflict and negotiation. *In Handbook of Organizational Communication: An Interdisciplinary Perspective*, pages 344–369, 1987.
5. T. R. Madanmohan, G. E. Kersten, S. J. Noronha, M. Kersten, and D. Cray. *Decision Support Systems for Sustainable Development. A Resource Book of Methods and Applications*, chapter Learning Negotiations with Web-based Systems. Kluwer Academic, 1999.
6. S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.
7. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

8. R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computing*, 29(12):1104–1113, 1980.
9. C. C. Koo. A commitment–based communication model for distributed office environments. In *proc. of Conference on Office Information System*, pages 291–298, New York, 1988. ACM Press.
10. J. L. Austin. *How to Do Things with Words.* Oxford University Press, Oxford, England, 1962.
11. J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge University Press, New York, 1969.
12. J.R. Searle. A taxonomy of illocutionary acts. In K. Gunderson, editor, *Language, Mind and Knowldge*, volume 7 of *Minnesota Studies in the Philosophy of Science*, pages 344–369, Minneapolis, 1975. University of Minnesota Press.
13. M. P. Singh. A Semantics for Speech Acts. *Annals of Mathematics and Artificial Intelligence*, 8(I–II):47–71, 1993.
14. R. Medina-Mora, T. Winograd, R. Flores, and F. Flores. The actionworkflow approach to workflow management technology. In *Computer-Supported Coooperative Work, Toronto*, November 1992.
15. P. Denning. Work is a closed loop process. In *American Scientist*, volume 80, pages 314–317, July-August 1992.
16. T. W. Malone, K. R. Grant K. Lai, R. Rao, and D. A. Rosenblitt. The Information Lens:An Intelligent System for Information Sharing and Coordination. *Technical Support for Work Group Collaboration*, pages 65–88, 1989.
17. M. K. Chang and C. C. Woo. A speech-Act-Based Negotiation Protocol : Design, Implementation and Test Use. *ACM Transactions on Information Systems*, 12(4):360–382, October 1994.
18. M. T. Jelassi and A. Forroughi. Negotiation support systems : An overview of design issues and existing sorftware. *Decison Support Systems*, 5:167–181, 1989.
19. E. Bellucci and J. Zeleznikow. A comparative study of negotiation decision support systems. In *Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98).* the IEEE Computer Society. Copyright (c) 1998 Institute of Electrical and Electronics Engineers, Inc., 1998.
20. F. Ackerman and C. Eden. Issues in computer and non-computer supported GDSSs. *Decision Support Systems*, 12(336):381–390, 1994.
21. I. Benbasat, F.J. Lim, and V.S. Rao. A framework for communication support in group work with special reference to negotiation systems. *Group Decision and Negotiation*, 4(371):135–158, 1995.
22. T. X. Bui and M. F. Shakun. Negotiation pocesses, evolutionary systems design, and NEGOTIATOR. *Group Decision and Negotiation*, 5(417):339–353, 1996.
23. J-M. Andreoli, D. Pagani, F. Pacull, and R. Pareschi. Multiparty negotiation for dynamic distributed object services. *Journal of Science of Computer Programming*, 31:179–203, 1998.
24. K. Benali, M. Munier, and C. Godart. Cooperation models in co-design. In *International Conference on Agile Manufacturing*, Minneapolis, USA, June 1998.