

# Network Support for Group Coordination

Hans-Peter Dommel

Computer Engineering Department, Santa Clara University  
Santa Clara, CA 95053, USA

and

J. J. Garcia-Luna-Aceves

Baskin School of Engineering  
Computer Engineering Department, University of California  
Santa Cruz, CA 95064, USA

## ABSTRACT

Recent advances in computer hardware and networking technology have incited the deployment of wide-area streaming media services in the Internet. While such efforts as video-on-demand are largely limited to unidirectional delivery of content to the desktop, synchronously interactive group-oriented application services are foreseeable. In such applications, users collaborate on a shared workspace and freely exchange information in real-time under the premise of coordination and conflict freedom. Telecollaborative applications such as telemedicine or distance learning may profit from such coordination services. Ultimately, group coordination allows for groupware-style computing at Internet scope. The current IP-multicast framework contains provisions for group membership control and reliable dissemination services, however, it lacks support for group coordination. In this paper, we present a framework on network control and coordination functions to orchestrate synchronous multimedia groupwork. Our goal is to achieve a better understanding of the group coordination problem as an important component of future Internet multimedia collaboration tools.

**Keywords:** Group Coordination, Distributed Resource Sharing, Network Support for Collaboration, Middleware.

## 1. INTRODUCTION

The proliferation of Internet services in recent years, in particular the World Wide Web, indicates the high demand for sharing of information through computer networks. A wider distribution of the work force, in form of telecommuting and ubiquitous computing [40], the advent of networked multimedia, and less expensive technology have shifted *telecollaboration* into the spotlight of mainstream computing. Telecollaboration comes in many faces, such as email, instant messaging, chat tools, application sharing, and real-time interaction on the same media or resources, qualified by the increasing degree of mutual awareness and the ability for instant information exchange and manipulation. Synchronous telecollaboration enables people in different geographic locations to share and jointly manipulate multimedia information in real-time and at various levels of granularity, bridging time and space. This aspect stands in contrast to legacy client-server applications such as Internet radio broadcast or video-on-demand, and to asynchronous, document-centric collaboration tools like email, instant messaging, or chat rooms. Representative application areas are collaborative virtual environments [6], distributed real-time gaming environments [7], distributed interactive simulations (DIS) [14], collaboratories [17], distance learning [23], and telemedicine [27].

Limitations in the availability and accessibility of resources in the shared workspace of a telecollaborative system create contention, competition, and conflict among users and make it necessary to deploy coordination mechanisms to reach consensus on how to jointly and effectively use the resources. Conflicts stalling the workflow may occur before and during resource allocation to users, as well as during actual usage. Telecollaborative services build on the provision of group coordination mechanisms. These manage access, manipulation, distribution and presentation issues between users and shared resources. Such coordination mechanisms are necessary to allow users to achieve individual goals in the context of group-centered remote interaction, when *telepresence* [3] substitutes for physical presence. Group coordination services support distributed hosts in coordinating their joint activities, to prevent or resolve resource contention, conflict and inconsistencies in the synchronous sharing of resources. Group coordination protocols, which embrace multicasting and consider network conditions in the coordination processes between hosts, complement efforts on group membership known from distributed systems and multicasting as an efficient message dissemination mechanism for group communication.

In this paper, we focus on network support for synchronous multimedia groupwork. We envision a new generation of collaborative multimedia systems using group coordination middleware to facilitate multipoint, multiparty, multichannel, and multimedia communication in small to very large groups and Internet scope. In such systems, groups and individuals can selectively, securely, and efficiently cocreate and disseminate information with improved telepresence and mutual awareness. The paper organization is as follows: Section 2 reviews related work. Section 3 discusses relevant components for network support of coordination services. Section 4 outlines key architectural issues in the design of group-coordinative systems. We conclude the paper in Section 5.

# Report Documentation Page

Form Approved  
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE <b>2006</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2006 to 00-00-2006</b>	
4. TITLE AND SUBTITLE <b>Network Support for Group Coordination</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>University of California at Santa Cruz, Department of Computer Engineering, Santa Cruz, CA, 95064</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES <b>6</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

## 2. BACKGROUND

Cerf *et al.* [5] pointed out the importance of transatlantic collaboration infrastructures in a memorandum in 1991. Our work is centered at the interface between legacy groupware [12], computer-mediated communication [25], and computer-supported cooperative work (CSCW) [31]. In their coordination theory framework, Malone and Crowston [21] define coordination as the “act of managing interdependencies between activities performed to achieve a goal”, looking at components of actors (people) and agents (computerized procedures), identifying workgoals, mapping goals to activities, and managing interdependencies among actors and activities. They distinguish between generic interdependencies, for instance sequenced or simultaneous actions on shared resources, and domain-specific interdependencies, e.g., specific data elements that must be passed between team members to achieve successful groupwork. Schmidt and Simone [34] present an empirical characterization of computational coordination mechanisms useful as general blueprint for the design of coordination protocols, proposing for instance the construction of a mechanism such that “actors are able to control its execution and make local and temporary modifications of its behavior to cope with unforeseen contingencies”.

Axelrod [1] investigated cooperation from a game-theoretic perspective, specifically, how the tradeoff between individual greed and good affects coordinative strategies in groups, assuming rational behavior. This problem is also known as the *social dilemma*. Focusing on “tit-for-tat” games, interactions are interpreted as pairwise alternations of moves with specific payoff values. The pay-off structure of the interaction determines the game motive. For two participants A and B, the payoff structure for choosing two actions  $i$  and  $j$  is  $P = A_{ij} + B_{ij}$ . If  $P = 0$ , then the interaction is called a *zero-sum* game, and interactions with  $P \neq 0$  are called cooperative or mixed-motive games.

A related approach uses economic models to tackle resource allocation in computer systems from a market-oriented perspective [15]. A cost function is assigned to cooperative activities, individual negotiations, deals, and strategies. An activity between two subjects is *pareto-optimal* if it is not possible to improve the utility for one subject without lowering the utility of the other subject. A strategy to determine the progress of activities is said to be in equilibrium if no party has an incentive to diverge from that strategy in order to fulfill individual and group tasks. Multiple equilibria are possible and two strategies  $S$  and  $T$  are said to be in *Nash-equilibrium* [24] if one party cannot do better other than using  $T$ , when the other party uses  $S$ , i.e., the product of individual utility values is maximized. However, it is not simple to assess global utility values, and choosing one of several possible equilibrium points may guarantee relative fairness but restricts the space of possible agreement states, under the assumption that all subjects employ the same utility measure and do not cheat. CONTRACTNET [36] was an early market-based protocol approach towards distributed task-completion, employing a bidding scheme among managing and contracting nodes. Shenker [35] argues that applying a fair-share service discipline at network switches models uncooperative flow control satisfying individual users’ selfishness more realistically than traditional disciplines, which presuppose cooperation such as First-Come-First-Served among users.

## 3. GROUP COORDINATION FRAMEWORK

We define coordination as an interactive scheduling process between two or more users forming a group to achieve joint work goals. Coordination correlates with cooperation, which we understand as the joint acting of individuals for a mutual benefit - in our context the mutual sharing of information for data mining or other forms of data exchange. We envision network-centric group coordination services to support distributed hosts in coordinating their joint activities, to prevent or resolve resource contention, conflict and inconsistencies in the synchronous sharing of resources. Key components for such services are the management of distributed resource access [8], ordered, reliable message dissemination [10], security [16], and stream synchronization [20]. Coordination and cooperation among users in networked multimedia systems support the process of *multimedia collaboration* [30], which is the actual act of users working together online.

We present a formal view on entities and actions refining earlier efforts [29, 30] on the definition of coordination and control processes in collaborative multimedia systems. Our work is process-oriented and differs from Candan *et al.* [4] who concentrate on algorithms for collaborative composition and transmission of media objects under given quality constraints, and their presentation in collaborative sessions.

We picture a computer network as a graph with nodes (stations, hosts)  $V$  sending messages across links (channels)  $E \subset V \times V$ . A connection is a unidirectional or bidirectional transmission link from a sender node to a set of receiver nodes.

DEFINITION 1. A *collaboration environment*  $\Gamma$  in a computer network is a tuple

$$\Gamma = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{F} \rangle \quad (1)$$

where  $\mathcal{S} = (V, E)$  is a set of sessions  $\Sigma$ ,  $\mathcal{U}$  is a set of users (hosts, processes, agents, participants),  $\mathcal{R}$  is a set of shared resources (media), and  $\mathcal{F}$  is a set of floors controlling the resources.

A session provides the infrastructure for coordination, cooperation and collaboration.

DEFINITION 2. A session  $\Sigma \in \mathcal{S}$  is a tuple

$$\Sigma = \langle Sid, T_i, T_e, A_S, L \rangle \quad (2)$$

where  $Sid$  is a unique identifier within  $\Gamma$ ,  $T_i$  is the initiation or announcement time,  $T_e$  is the ending time, and  $A_S$  is a list of attributes characterizing the session at level  $L$ . A conference is a set of sessions  $\Sigma_i \in \mathcal{S}$ , where  $i \geq 1$ .

$Sid$  is a unique session identifier per collaborative environment, whose sequence number space is wrapped around in correlation with the turnover rate and lifetime of sessions in  $\Gamma$ . The time may reflect real-time, logical time, or define a lifetime interval  $\Delta = T_e - T_i$ .  $L$  denotes the session level (default 0).  $A_S = (M, O, C)$  describes purpose and orchestration of a session in terms of membership  $M$ , organization  $O$ , and control  $C$ . Sessions can be flat ( $L = 1$ ) or maintain two or more levels with *nested* groups ( $L > 1$ ).

Szyperski [39] characterizes session types in a similar, but less refined way, according to the model of interaction (controlled, dynamic, static) and data flow ( $1 - n, n - 1, m - n$ ). For instance, a lecture is a controlled, long-term interaction between one sender and  $n$  receivers. Telemetry is a typical  $n - 1$  session, and a whiteboard session is typically  $m - n$ . Our session characterization applies to specific collaborative applications, as well as generic session types in the spectrum of real-time collaborative work, such as lectures, business meetings, labs, panels, brainstorm meetings, exams, interviews, or chats.

*Membership* ( $M$ ) reflects the composure of the user group in the session. *Participation* specifies whether information is exchanged unilaterally, or bilaterally relative to a host, impacting user access rights and data-flow. Interactive sessions may be symmetric, i.e., all users have the same view on shared resources (WYSIWIS), or asymmetric, where users pertain individual views on the same shared data space (relaxed WYSIWIS) [37]. *Size* specifies a small ( $< 5$ ), medium ( $< 100$ ), or large ( $\geq 100$ ) number of users, impacting scalability of the coordination mechanism. *Accessibility* declares whether a session is open, allowing any user to join, whereas closed sessions allow participation by invitation only. *Authorization* specifies whether coordination primitives may use read-only, read-write, or write-only privileges for the entire session. Users may have individual, role-based authorizations, as well.

*Organization* ( $O$ ) entails specifics on how the session is to be orchestrated. *Dataflow* describes how data are multiplexed among users, with a  $1 - 1, 1 - n$ , or  $1 - m$  transmission model and with unicast, broadcast, or multicast in a session of  $n$  users, where  $m \leq n$ . Delivery can be ordered or unordered. *Duration* discerns between sessions with longer lifetime (persistent) vs. short-term sessions, where the precise timing modalities are case-specific and left open. The *scope* specifies the hop limit for packets sent by hosts in a particular session, similar to the Time-To-Live semantics in IP, which allows constraining sessions to a geographic range and retain privacy or limited dissemination to a specific group. *Media composition* defines whether the session uses a single medium such as audio-only, or mixed media, e.g., a video-audio combination. *Conduction* refers to the session agenda and moderation style, which can be either tightly coupled, i.e., all users know about each other and follow some agenda in the style of “Robert’s Rules of Order” [32], or the exchange is loosely-coupled and not prescribed.

*Control* ( $C$ ) depicts the status, locus of control, and security measures activated for a session. Sessions with overlapping or diverging interests can merge or split. Such reconfiguration of sessions with regard to membership and session events linked to specific phases must be possible without session termination or restart of applications. The session *status* marks whether the session is a partition from a larger session, frozen but still deemed as active, merged or revived. Tracking of states in coordination protocols and the outcome of coordination processes can be logged and persistent, or ephemeral.

*Locus of control* specifies, whether membership and floor control are being handled in one central location, partially distributed among several servers, or fully distributed across all hosts. Partial or full replication is possible for the latter two paradigms. A central controller can also rove among all sites and achieve better fault tolerance. Distributed control is multilateral, with varying degrees of “consentience” and “equipollence”, i.e., how much everybody participates and how authorities and responsibilities are allocated. Multilateral control is either successive, partitioned, democratic or anarchic. Successive controllership allows one distinct controller at a time, and alternates among users, and partitioned control lets several controllers each perform a subset of control operations. Democratic control lets all users contribute to the control process, e.g., via voting. Anarchic control gives all subjects complete freedom of acting and control of sharing is peer-to-peer based.

The control locus is related to the *supervision* attribute, indicating whether the communication process in coordination is moderated, peer-reviewed, or free. A moderator decides which users may send information, what is forwarded to the receivers, or which receivers may receive a particular content or access a specific resource, implementing a notion of floor control. McKinlay *et al.* [22] note for face-to-face meetings that the importance of chaired guidance increases with the session size, and the difficulty in performing a joint task, since each member’s ability to participate and influence others is reduced. Finally, coordination touches upon *security* issues, specifying whether users are anonymous or authenticated in their exchanges, either at session initiation, or at every turn, and whether information is encrypted. Rajan *et al.* [29] identify a *confluence* as a special session type, where all participants transmit and receive the same set of media streams mixed together

in broadcast, which saves bandwidth. The notion of confluences and session nesting leads to the concept of multilevel or *hierarchical* sessions, whose discussion we omit for space reasons.

DEFINITION 3. A user  $U \in \mathcal{U}$  is a tuple

$$U = \langle Uid, Sid, Loc, T_j, T_l, A_U \rangle \quad (3)$$

where  $Uid$  is a unique identifier within the session  $Sid$ ,  $Loc$  is the local or remote location, given as IP-address or unique host identifier,  $T_j$  is the joining time,  $T_l$  is the leaving time, and  $A_U$  is a list of user attributes.

Users can be represented by system agents [11, 19]. Accordingly, users are characterized by their roles, authority, identity, entry capabilities and access rights, which impact the applicable floor control strategy. Users can be co-located in the same space, or geographically distributed. We distinguish between social and system *roles*. *Social roles* describe the function of a user within a session, e.g., being a panelist or lecturer. *System roles* refer to the control function within a floor control protocol: participants without a specific role can be either a receiver or inactive. The owner of a resource  $R$  is the node that injects  $R$  into a session and initiates floor control for  $R$ , which may vanish from a session if the owner leaves. The floor coordinator ( $FC$ ) is an arbiter over a resource  $R$ , or a session moderator granting or denying a floor on  $R$  during session time to the floor holder ( $FH$ ), who attains the exclusive right to work on  $R$  for a floor holding period.  $FC$  and  $FH$  may be located at different nodes, or be assumed by the same node. These roles may be statically assigned at session start, or rove among users during session conduction. Users without control roles are general session members, and can be active or inactive, depending on whether they invoke state transitions in the coordination mechanism. Role-based floor control in dynamic sessions contrasts static *role-based access control* (RBAC) models [33].

In the list of user attributes  $A_U$ , *Authority* defines whether the user is a simple participant, privileged as system root user, or moderator, linking this field with the role entries. A moderator can be permanent  $FC$ . As a social role, the moderator equates to a session supervisor being able to inspect all session turns between users. *Identity* specifies whether the user wants to remain anonymous or whether the  $Uid$  can be posted to the session. An *entry* is either independent, i.e., unaware of the actions and entries of others, reflective, i.e., polling session members, consultative based on “contextual clue messages”, partitioned and representing a subtask, based on voting among the group, or debriefed and recorded [12]. In addition, user entries may be temporary or permanent, and logged for the purpose of reviewing histories of collaborative sessions, or for undoing certain steps [28]. *Access* defines the basic privileges for working on a resource, in receive-only, send-and-receive, and send-only mode, analogous to read and write authorizations in file systems. Aggregation of users leads to the notion of groups:

DEFINITION 4. A user group (multicast group)  $G$  is a set of users  $U$  with common session and user attributes, expressing a common media or task focus, such that  $\mathcal{U} \supseteq G \supseteq U$ .

DEFINITION 5. A resource  $R \in \mathcal{R}$  is a tuple

$$R = \langle Rid, Sid, Pid, Uid, T_c, T_d, A_R \rangle \quad (4)$$

where  $Rid$  is a unique resource identifier owned by user  $Uid$  within session  $Sid$ .  $Pid$  is the parent identifier or the resource that  $Rid$  belongs to,  $T_c$  is the time of creation or injection of the resource into the collaborative workspace,  $T_d$  is the deletion time, and  $A_R$  is a list of resource attributes.

$Rid$  designates both discrete media and streaming media and may contain the port where the resource is transmitted. The  $Pid$  value allows for recursive subsumption of resource components within resources, and hence sharing or resource components at an arbitrary granularity. For instance, users can share an entire window, or a graphical object within that window. Among the relevant resource attributes  $A_R$ , *Class* describes whether the resource is continuous or discrete. *Type* characterizes the media object class, indicating whether a resource is text-based, graphical, or some real-time medium and identifies the purpose it serves. *Usage* determines if the resource can be used concurrently by multiple users or requires sequential processing with exclusive floors. For instance, a shared whiteboard allows for multiple concurrent telepointers with a small number of users, whereas a remotely controlled camera can only perform a positioning command for one user at a time. *Priority* sets an importance value on the transmission and processing of the information, preempting other media dissemination of lower ratings. *QoS* defines the required Quality-of-Service [38] for the resource, including the tolerable loss, the required resolution, the possible maximum delay, and the color depth. Other criteria may be added depending on the nature of the resource, such as the channel number, a frame-rate, encoding scheme, sampling rate etc. The *Protection* attributes denotes whether a resource is public, private, or proctored, which may be expressed with a numerical value, or work in analogy with the *Bell-LaPadula* model [2], discerning between top-secret, secret, confidential, or unclassified information [13]. The last component describes access privileges to the collaborative workspace, called “floors”:

DEFINITION 6. A floor  $F \in \mathcal{F}$  is a tuple

$$F = \langle Fid, Rid, Uid, T_i, T_d, A_F \rangle \quad (5)$$

where  $Fid$  is a unique floor identifier within the shared workspace for a resource  $Rid$ , assigned to user  $Uid$  at inception time  $T_i$ , and deactivated at time  $T_d$ , with  $A_F$  denoting a list of floor attributes.

Note that one  $Rid$  may have multiple  $Fid$  assigned for control of various granules, but each floor is controlling exactly one resource. Floors are indirectly associated with sessions via  $Rid$ , and floor properties may be inherited from a master resource to its subcomponents. We assume that one floor  $F$  is assigned per resource component. The pairing  $(Fid, Rid)$  specifies the granularity of control and the commands available with possession of the floor. A floor can control an entire conference, an application, a single window, or a shared object [18]. For instance, for audio the associated commands may be `talk`, `mute`, `pause`. Video floor commands are for instance `caption`, `forward`, `cut`, `replay`. Floors can be static relative to a session lifetime, or dynamic, i.e., assigned ad hoc by a computer or social protocol. The combination of  $Uid$  and the attributes specifies whether the user is  $FC$ ,  $FH$ , chair, or general participant.  $T_i$  and  $T_d$  may be set using real-time clocks, or a logical session time. The floor attributes  $A_F$  comprise directionality of control, state, instantiation, passing rules, connection modality and access strategy.

The presented model serves both theoretical and practical purposes. It provides a more elaborate framework for formal specification and validation of collaborative systems, e.g., with the prototype verification system [29]. It also allows for session capability descriptions [26] to set up and query the membership and coordination status of an active conference, where a capability is understood as a resources or system feature influencing the selection of useful configurations for components. We have also developed an activity semantics describing causality and coordination constraints using the presented taxonomy. It is not our intention to provide a comprehensive model and parameterization of coordination services, but rather discuss key concepts as a stepping stone toward more sophisticated design and deployment of software using group coordination.

## 4. SYSTEM ARCHITECTURE

In contrast to the majority of commercial and experimental CME existing to date, we look at collaboration as an inherently distributed process, where session coordination and control are enacted collectively by participating hosts, rather than fixing such roles in centralized servers. We postulate henceforth a coordination architecture with the following requirements: simplicity of implementation and maintenance; scalability in the number of users and hosts; security with regard to the exchange of coordination information and data; extensibility for new resources and session models; efficiency in coordination, concerning low latency and protocol state overhead; reliability with regard to failures of hosts, resources, and network elements; persistence of coordination information at hosts despite the ephemeral nature of access permission exchanged between collaborating sites; and interoperability between heterogeneous platforms. A more elaborate view on this architecture is presented in [9].

## 5. CONCLUSION

A comprehensive framework for group coordination in networked multimedia systems has been presented. The framework has its foundation in a formal model of group coordination and collaboration, related to hierarchical session control and role-based session participation, revolving around the notion of turn-taking in interactive groupwork. Important design issues for such an architecture have been discussed in conjunction with the various media and session types and their properties. Our coordination model does not represent a panacea for the many open problems encountered in groupware, CSCW and networked multimedia systems. Rather, we intend it to be an integrative step towards a better understanding of group collaboration, and more flexible, rich middleware services to facilitate it.

## ACKNOWLEDGMENTS

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under grant F19628-96-C-0038.

## REFERENCES

- [1] R. Axelrod. *The Evolution of Cooperation*. Basic Books, Inc., New York, NY, 1984.
- [2] D. E. Bell and L. J. La Padula. Secure computer system: Unified exposition and multics interpretation. Technical Report MTR-2997, Mitre Corp., Bedford, MA., July 1975.
- [3] G. Bell. Telepresence. Presentation at UC/Berkeley, Feb. 1996.
- [4] K. S. Candan, V. S. Subrahmanian, and P. V. Rangan. Towards a theory of collaborative multimedia. In *Proc. 3rd IEEE Int. Conference on Multimedia Computing and Systems*, pages 279–282, Hiroshima, Japan, June 1996.
- [5] V. Cerf, P. T. Kirstein, and B. Randell. Network and infrastructure user requirements for transatlantic research collaboration. RFC 1210, March 1991.
- [6] E. F. Churchill and D. Snowdon. Collaborative virtual environments: an introductory review of issues and systems. *Virtual Reality*, 3(1):3–15, 1998.

- [7] C. Diot and L. Gautier. A distributed architecture for multiplayer interactive applications on the Internet. *IEEE Network*, 13(4):6–15, July-Aug. 1999.
- [8] H.-P. Dommel and J. J. Garcia-Luna-Aceves. Floor control for multimedia conferencing and collaboration. *Multimedia Systems J. (ACM/Springer)*, 5(1):23–38, Jan. 1997.
- [9] H.-P. Dommel and J. J. Garcia-Luna-Aceves. A coordination architecture for Internet groupwork. In *Proc. 26th EUROMICRO Conference - Informatics: Inventing the Future, Workshop on Multimedia and Telecommunications*, Maastricht, Netherlands, Sept. 2000. IEEE.
- [10] H.-P. Dommel and J. J. Garcia-Luna-Aceves. Ordered end-to-end multicast for distributed multimedia systems. In *Proc. 33rd Hawaii Int. Conf. on System Sciences*, Maui, Hawaii, Jan. 2000.
- [11] E. A. Edmonds, L. Candy, R. Jones, and B. Soufi. Support for collaborative design: Agents and emergence. *Comm. of the ACM*, 37(7):41–47, July 1994.
- [12] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware - some issues and experiences. *Comm. of the ACM*, 34(1):38–58, Jan. 1991.
- [13] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems, 2nd Ed.* The Benjamin/Cummings Publ. Company, Redwood City et. al., 1994.
- [14] Inst. Electr. & Electron. Eng. IEEE standard for Distributed Interactive Simulation - Application protocols. Report, New York, NY, Aug. 1998.
- [15] D. F. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini. Economic models for allocating resources in computer systems. In *S.H. Clearwater (Ed.) - Market-based control: a paradigm for distributed resource allocation*, pages 156–183. World Scientific, 1996.
- [16] L. Gong. Enclaves: Enabling secure collaboration over the Internet. *IEEE Journal on Selected Areas in Communications*, 15(3):567–575, April 1997.
- [17] R. T. Kouzes, J. D. Myers, and W. A. Wulf. Collaboratories: doing science on the Internet. *Computer*, 29(8):40–46, Aug. 1996.
- [18] J. C. Lauwers and K. A. Lantz. Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems. In *Proc. SIGCHI*, pages 303–311, Seattle, WA, Apr. 1990.
- [19] K.-C. Lee, W. H. Mansfield Jr., and A. P. Sheth. A framework of controlling cooperative agents. *IEEE Computer*, pages 8–16, July 1993.
- [20] W. Liao and V. O. Li. Synchronization of distributed multimedia systems with user interaction. *Multimedia Systems*, 6(3):196–205, May 1998.
- [21] T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, March 1994.
- [22] A. McKinlay, R. Procter, O. Masting, R. Woodburn, and J. Arnott. Studies of turn-taking in computer-mediated communications. *Interacting with Computers*, 6(2):151–171, June 1994.
- [23] C. D. Miller and R. W. Clouse. Technology-based distance learning: present and future directions in business and education. *J. of Educational Technology Systems*, 22(3):191–204, 1993-1994.
- [24] J. F. Nash. The bargaining problem. *Econometrica*, 28:155–62, 1950.
- [25] D. G. Novick and J. Walpole. Enhancing the efficiency of multiparty interaction through computer mediation. *Interacting with computers*, 2(2):227–246, Aug. 1990.
- [26] J. Ott, D. Kutscher, and C. Bormann. Capability description for group cooperation. Internet Draft draft-ott-mmusic-cap-00.txt, June 1999.
- [27] V. L. Patel, D. R. Kaufman, V. G. Allen, E. H. Shortliffe, J. J. Cimino, and R. A. Greenes. Toward a framework for computer-mediated collaborative design in medical informatics. *Methods of Information in Medicine*, 38(3):158–176, Sep. 1999.
- [28] A. Prakash and M. J. Knister. A framework for undoing actions in collaborative systems. *ACM Trans. Computer-Human Interaction*, 1(4):295–330, Dec. 1994.
- [29] S. Rajan, P. V. Rangan, and H. M. Vin. A formal basis for structured multimedia collaboration. In *Proc. 2nd IEEE Conf. on Multimedia Computing and Systems*, Wash., D.C., May 1995.
- [30] P. V. Rangan and H. M. Vin. Multimedia collaboration as a universal paradigm for collaboration. In *Multimedia - Principles, Systems and Applications*, pages 3–15. Springer-Verlag, Apr. 1991.
- [31] W. Reinhard, J. Schweitzer, and G. Völksen. CSCW tools: Concepts and architectures. *Computer*, pages 28–36, May 1994.
- [32] H. M. Robert. *Robert's rules of order*. Bantam Books, Toronto; New York, 1986.
- [33] R. S. Sandhu and E. J. Coyne. Role-based access control models. *Computer*, 29(2):38–47, Feb. 1996.
- [34] K. Schmidt and C. Simone. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer-Supported Cooperative Work*, 5(2-3):155–200, 1996.
- [35] S. J. Shenker. Making greed work: A game-theoretic analysis of switch service disciplines. *IEEE Trans. on Networking*, 3(6):819–831, Dec. 1995.
- [36] R. G. Smith. The ContractNet protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. on Computers*, 29(12):1104–1113, Dec. 1980.
- [37] M. Stefik, G. Foster, D. G. Brobrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Comm. ACM*, 30(1):32–47, Jan. 1987.
- [38] R. Steinmetz and K. Nahrstedt. *Multimedia: Computing, Communication, and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.
- [39] C. Szyperski and G. Ventre. A characterization of multi-party interactive multimedia applications. Technical Report TR-93-006, International Computer Science Institute, Berkeley, Feb. 1993.
- [40] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):74–84, July 1993.