Design issues for floor control protocols

Hans-Peter Dommel and J.J. Garcia-Luna-Aceves
Baskin Center for Computer Engineering & Information Sciences
University of California
Santa Cruz, CA 95064
`peter@cse.ucsc.edu` • `jj@cse.ucsc.edu`

ABSTRACT

Floor control allows users of networked multimedia applications to remotely share resources like cursors, data views, video and audio channels, or entire applications without access conflicts. Floors are mutually exclusive permissions, granted dynamically to collaborating users, mitigating race conditions and guaranteeing fair and deadlock-free resource access. Although floor control is an early concept within computer-supported cooperative work, no framework exists and current floor control mechanisms are often limited to simple objects. While small-scale collaboration can be facilitated by social conventions, the importance of floors becomes evident for large-scale application sharing and teleconferencing orchestration.

In this paper, the concept of a scalable session protocol is enhanced with floor control. Characteristics of collaborative environments are discussed, and session and floor control are discerned. The system's and user's requirements perspectives are discussed, including distributed storage policies, packet structure and user-interface design for floor presentation, manipulation, and triggering conditions for floor migration. Interaction stages between users, and scenarios of participant withdrawal, late joins, and establishment of subgroups are elicited with respect to floor generation, bookkeeping, and passing. An API is proposed to standardize and integrate floor control among shared applications. Finally, a concise classification for existing systems with a notion of floor control is introduced.

Keywords: computer supported cooperative work, collaborative multimedia computing, concurrency control of shared multimedia objects (floor control), media and user interaction.

1 INTRODUCTION

Communication in the real world is based on sharing the same space like a conference room, the same ether for sound exchange and other media resources. Likewise, in a Computer Supported Cooperative Work (CSCW) setting, people share a network of machines, the same applications and media, to facilitate remote interaction, data-sharing and interactive collaboration. For teleconferencing such multiparty interaction for exchanging multimedia-data has to be facilitated under real-time constraints. Remote communication is different in that implicit social conventions based on deictic and mimic gesture due to lack of personal presence cannot be employed fully.[16] In the last decades, efforts have been made to let hardware platforms feature *interoperability* and software to feature *compatibility*. This is now being enhanced by introducing *collaborability* between application processes.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **1995**	2. REPORT TYPE	3. DATES COVERED **00-00-1995 to 00-00-1995**

4. TITLE AND SUBTITLE **Design issues for floor control protocols**	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) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	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

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **12**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

For multimedia systems, the notion of collaboration changed from simple text-based "chatting tools" or testbeds for workgrouping with teleconferencing equipment to a broad spectrum of *groupware*, integrating software for textual, visual, and auditory communication.

Because face-to-face meetings are replaced in a Computer Mediated Communication (CMC) setting by conferencing tools, a supplementary service is needed to coordinate multiple usage of sharable data among users from distinct remote sites. The concept of *floor control* aims at providing such orchestration service in order to guarantee that at any given moment only a selection of users is allowed to simultaneously work with or on the same shared objects, creating a virtual and temporary exclusivity for access on such objects. Floor control performs hence controlled access of subjects (users) to shared objects (resources). Resembling traditional concurrency control techniques used for database systems or static file permission schemes in operating systems, floor control aids or replaces a conference chair in order to model *turn-taking* behavior in collaborative activities and avoid race-conditions or conflicts, hence making groupwork more effective. Floor control design issues, encompassing protocol and information handling, and the user-interface, will be discussed in the following.

2 COLLABORATIVE ENVIRONMENTS

Current collaborative environments feature a variety of software for communicating and collaboration, e.g., for work with *text* (chat-tools, mail, spreadsheets, editors, hypertext browsers), *sound* (voice, music), *images* (still and motion video, facsimile), *graphics* (whiteboard), and *dedicated shared applications* (visualization, scheduling, decision support, workflow systems etc.). For floor control services, the following characteristics of a collaboration environment are decisive:

- *Distributed* state information storage for *scalability*[39] of session and floors, *performance* (reduced network traffic, especially with multipoint connections), *graceful degradation* in congestion situations, and *fault tolerance* in case of site-crashes (vs. a *centralized* approach, where the server is a single failure point and bottleneck for responsiveness).
- *Asymmetric* interaction, i.e., conferees need not have the same software state and data, but rather work *asynchronously* in a *heterogeneous* setting with mixed media and share, when appropriate (vs. a *symmetric* and *synchronous* model, where replicated software produces the same output, not allowing for independent local processing).
- *Hierarchical* sessions, being tree-structured into subsessions or subgroups (*coteries*) to handle side-chats etc. without requiring establishment of separate sessions. Temporary floors need to be created for the coterie resources, inheriting attributes from the parent session[37] (vs. a *flat* session model without refined group support).
- *Tightly-controlled* conferencing, i.e., complete information on the floor states is shared and consistently maintained by conferees (vs. a *loosely-controlled* model, where conferees tune into a conference via an agreed-upon multicast address, and a conference state is reached asynchronously via passive reception of control messages without direct end-to-end coordination).

The Quality-of-Service (QoS) of a floor control architecture is reflected in the acceptance of such a mechanism by the audience. Criteria are *correctness*, *fairness* and *promptness* of floor assignment, *adaptability* to various resource and collaboration scenarios, based on network parameters like traffic (throughput, burstiness, delay and jitter), *synchronization* control between different media, and *reliability* (loss probabilities and fault tolerance).

3 SESSION CONTROL AND FLOOR CONTROL

Managing collaborative efforts among remote sites requires services for session management, floor control, authentication, synchronization among mixed media, etc. We focus on the first two and briefly characterize them:

- *Session control*[10] designates meeting coordination between a changing set of remotely located users based on a connection management protocol. It mediates between upper application layers and relays requests down to end-to-end services. Users are supported in establishing a session, joining a running session, withdrawing from a session, partially in specific resources, temporarily, or completely, or in inviting to a conference as participants or "third-party" bystanders.
- *Floor control* designates keeping track of the users' usage of media channels and shared resources, orchestrating *mutually exclusive* resource access between users *safely* (a floor is assigned correctly to requesting users), *reliably* (control information transmission works without packet loss), and in *real-time*. To account for mixed media, a floor control protocol must be *generic* (encompass any type of application and shared objects) and *adaptive* (accommodate heterogeneous software configurations on collaborating sites). For user acceptance, floor assignment must be *fair* (no user "starves"), *intuitively correct* (close to the "look-and-feel" of a face-to-face meeting), and *non-intrusive* (leaving freedom for conferees to regulate floors manually).

Both services can be integrated into a *collaboration-aware*[24] application or they can be extracted into *daemons* running independently on every site and mediating between groupware events, concentrating all connection and floor distribution related efforts into distinct agents and hence avoiding duplication of such efforts in the applications themselves.

4 DESIGN DECISIONS FOR FLOOR CONTROL SERVICES

Design of floor control services addresses system issues (packet structure for the protocol, distribution policies, and exceptions like system crashes), and user interface issues (how floors as virtual permission attributes are presented to a conferee and how their distribution can be triggered and manipulated). Both perspectives are shortly elicited.

4.1 System related design

There are various ways to implement a floor control scheme and the one favored here is based on the concept of an agent that runs like a daemon process on every site involved. All incoming and outgoing requests and data run through this floor control unit. Such a setup allows for channeling all floor requests and centralizing them in one local process instead of having many independent requests floating between involved sites that are hard to coordinate. It also allows for switching floor management for individual applications and media on or off. A distributed floor control algorithm decides for all connected machines and respective media, which participant at a given time is eligible to certain actions on a particular shared resource.

Protocol design can be based on traditional concurrency control due to inherent similarities of managing shared resources, with floors as dynamic write-permissions. However, multimedia sharing is not based on *user–resource transactions*, but rather on *user–medium–user interactions*. Also, bandwidth, delay and reliability requirements are different, and most media, e.g., voice channels, do not allow for rollbacks.

Two participants define the minimum setup for a collaboration. Connections can encompass different distribution ranges from local to worldwide. Conference sessions can be characterized in the *number of participants*, the *organizational structure* (chair-guided, nonhierarchical, hierarchical), the *purpose* (lecture, seminar, casual meeting, planned meeting, public hearing, panel discussion etc.), the *granularity* of collaborators (single, subgroup, group), and their *interconnectivity* (1-1, 1-n, m-n). Each conferee has a *role*, namely as meeting host or initiator, floor controller (maintaining information for one specific floor), current holder (being granted the floor), or participant (aspiring to be holder or controller). Participants are generally humans, but can also be agents or other addressable network entities present in the course of a conference. Participants can have multiple roles, i.e., being initiator, chair, or receiving or sending participant at the same time. Floors can be granted individually for any type of application with varying granularity (cursor, event, window, file, media channel, etc.), allowing for single or multiple access of conferees to the critical section of information sharing for public data objects. A floor is attributed through a sequence of events on different abstraction levels, as depicted by the simplified *causal chain* in Figure 1.

Site -> Session -> Group -> Subgroup -> User -> Role -> Application -> Shared resource -> *Floor*

Figure 1: Interdependency of collaborative parameters for determining floor parameters (multiple arrows indicate several possible choices).

Floor control information, encoded in packets transmitted between collaborators to gain consensus on the current *floor holder* for a specific medium, is characterized in Figure 2.

SessionId	**Hostname**	**GroupId**	**CollaboratorId**	**Role**	**Application**	**ResourceType**	**FloorId**	**FloorState**

Figure 2: Packet structure for Floor Control Information.

The Role refers to whether a user is a floor holder (current manipulator of a resource), controller (maintaining information in the floor), or conferee without floor rights. It can be refined to allow for a "nice-value", similar to process priority settings in UNIX, to enforce preferenced floor attribution to specific task-holders. The GroupId can encode a subgroup, allowing for multiple concurrent floors between pairs of conferees, if their concurrent work on the same medium (e.g., a voice channel) does not overlap. The ResourceType refers to the underlying media (e.g., voice or text-based) which require different floor distribution policies. The FloorState (e.g., `granted`, `requested`, or `free`), must be further specified within an actual protocol. The above packet structure yields unique identification of each floor for any participant, application and current action. Packets with varying control information composed from this floor tuple will precede data packets with information on shared events in order to update the floor states on cooperating sites.

A specific floor is wandering between those sites with the same software, sharing the adjunct resource, and floor distribution is handled between those sites. In order to guarantee consistency among all collaborating sites with regard to specific media and their floors, floor state tables need to be synchronously updated after each floor state change. The actual floor *policy* (e.g., free-for-all, chair-designated, first-come-first-served, least-recently-served, round-robin, contention-avoidance, leader-election etc.) and the number of concurrent floors may vary in the protocol, depending on the shared resource target, whereas the distribution *mechanism* across all sites remains the same. For instance, in a 1-to-1 conversation, a voice floor needs to be granted to both sites to allow for rapid changes in the speaker role of a dialogue, whereas a large session with one speaker in a monologue and many listeners requires only one floor holder.

Delays in message passing due to system performance and end-to-end service limitations cause *grey zones* in floor passing periods. Such "floor outages" can be resolved by assuming that one floor is attributed to one site, until the turnover site is acknowledging explicitly the reception of the floor. Also, in order to provide a facility for shortterm interjections in collaboration, e.g., for voice-based telepresence, a *backchannel floor* needs to be provided at every site as the interactive counterpart to the main *action floor* of the current holder, to establish a truly bidirectional cooperation and account for interruptions by "out-of-turn-speakers", cause *delayed completion* of the floor holder's action.[25] Twofold floor-establishment must not be restricted to voice-channel mediation, but can be applied to any media, which supports feedback in cooperative behavior, e.g., in a whiteboard for small additions or corrections. Figure 3 displays the conversational process in a two-point connection and the "flip-flop pattern" of primary and secondary (feedback) floor. The turnover times, depicted here as ideal sharp transitions, incur some delay in reality.

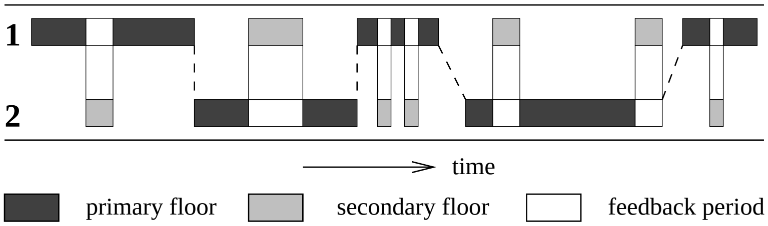

Figure 3: Two floors as counterparts allowing for interjections in a collaboration.

Although each media tool has its own transport service, floor control information is bundled for all media types into one generic packet structure. A reliable multicast protocol on top of unreliable network service is a commonly proposed compromise in the current literature. Floor control state information can be incorporated in packet headers of resource reservation protocols, e.g., in order to observe real-time deadlines.

4.2 User related design

Although a floor is essentially a virtual attribute, it must be visible and modifiable to a collaboratee through the user-interface – in windows and views, the form or color of the shared object, and in some menu showing who has the floor on which resource. Similar to access control lists as protection and capability structures,[40] more granularity in assignment of specific permissions can be achieved by maintaining a selection of eligible recipients for a specific floor or related data in pull-down menus. Default floor attribution must hence be overridable by individual user settings. Essential user-related criteria are, how floors and state information are presented in the interface, how floor passing among users is triggered, and how *coherence* between data located and displayed at remote sites is maintained.

For the *presentation of floors and state information* private windows and widgets hold non-sharable data and public windows display data that are selectively shared. Only public resources are involved in floor control and their visual representations via icons, widgets etc. serve as floor access "keys". *Local resources* originate from the local site and are transmitted elsewhere, and *remote resources* originate on remote machines and are displayed locally. Per resource there is one floor, owned by the user who created the resource and associated data. Floors can be attached with varying granularity, reaching from a shared cursor to a viewing window, an intelligent tool with menu-controlled functions, or an entire application. We distinguish between *sender floors* and *receiver floors*. Sender floors are explicitly granted to collaborators for manipulation and display of their data on remote sites (*What You See Is What I Share*). Receiver floors filter incoming data from other sites on remote shared resources, reducing the onscreen information load (*What I See Is What I Want*). Receiver floors are only set locally and do not affect remote work processes. Receiver and sender floor control runs concurrently on every site to check both for objects handled locally and remotely.

Floor presence can be coupled with window control, i.e., iconization of a window disables holding of a floor. For instance, a remote user might be still speaking, and have the voice floor, but is set mute. Further salient features are the alignment of related information, participant autonomy, the distinction between shared and private spaces, the separation of conference-control and application-related commands, the agreement upon conference roles for participants and a continuous presentation of the conference status. A list of active participants can for example be provided in a pull-down menu attached to each shared resource, allowing to specify sender and receiver floors. Collaborations via certain media channels or on specific resources can be rendered graphically as clusters of user icons and pointing at a cluster would mean to share data with that group and their agreed-upon resources.

Since data and floor agents are distributed, the resource interface need to be locked, when in request-state, otherwise runaway-floors for subsequent requests are possible. Visual cues like transparency or color in the representation of a shared resource are suggested to indicate its current floor state, e.g., an *opaque* or green rendering can depict a locally held floor, a *light-shaded* or yellow rendering indicates that a floor is requested by a remote site, and a *transparent* or red object icon indicates that floor is held by a remote site. Auditory cues can also be used selectively, i.e., pointing on a locked resource could depict the status of sharing via an auditive signal.

For the *triggering of floor passing* various control mechanisms are possible to indicate relinquishing or grabbing of a floor, e.g.,

1. *chair-guided* control, where one conferee serves as facilitator of collaborative actions and assigns the floor "manually ";
2. *cue-triggered* control, e.g., *mouse-triggered*, *voice-activated*, or *gesture-based* via a data-glove. For instance, pressing a mouse-button (or key) to claim a floor, while pointing at a shared resource, is simple and intuitive, but may be problematic, if a user temporarily releases the button without intending to release the attached floor;
3. *unbound* control, depending on agreements between users on when to revoke or relinquish a floor – a possibly unfair solution, since some users might never release a floor;
4. *time-bound* control which restricts floor-holding, e.g., to a default time-limit or the accumulation of a certain number of requests from remote sites – a possibly unacceptable solution, since users might feel not in control and rushed.

Floor passing may not depend solely on the (non)activity of a resource. A queuing scheme may account for delays in transport of requests, but at the same time cause loss of real-time QoS. 1) – 3) exemplify *explicit* floor control, triggered directly by humans, whereas 4) represents *implicit* floor control, triggered by some automatic mechanism, which is favorable for certain meeting purposes or media. We favor explicit floor control and assume that race-conditions for floor requests and releases can be negotiated between users, especially if a "fast" medium like voice is involved. *Responsiveness* is essential for good QoS, to yield appropriate *response time* (the local processing time for a collaborative action and its reflection in the user interface), and *notification time* (the time needed to propagate local actions to conferees' sites and interfaces).[11] Finally, the floor mechanism must preserve the *atomicity* of a user's action on some resource, i.e., floor requests should not imply automatic floor revoking. This allows for a notion of *commits* on collaborative actions and the *coherence* of distributed data, as well as, on demand, of interfaces and displayed data. Such consistency among sites must not be preserved at any time, but can adhere to a model of *lazy consistency*[29] with incremental updates, only when a specific resource and link are actually activated. Also, for visible media, e.g., text-editors or whiteboards, a *history* and *undo*[36] service have to be provided.

4.3 Stages of interaction

We can identify three timely stages in collaboration, which affect floor generation, bookkeeping, and passing:

- *Initiation:* Floors are created with the establishment or joining of a session. Additional resources can enter the sharing-pool later, introducing one new floor per resource at the site of the creator, who is also the initial holder of the resource.

- *Flow:* Control information transmissions need to be point-to-point and reliable. Only the first request arriving in a sequence is accepted and the follow-ups are discarded, similar to a conference, where a participant has to raise a hand or voice several times to be heard, or consecutive requests are buffered and processed according to the chosen policy. Floor information is changed as soon as the state of the floor holder changes, i.e., if he or she relinquishes the floor etc. Special cases are withdrawing, returning, joining late (invited or uninvited) and on-the-side collaborations in subgroups. For late joins, an update of the latest floor state must be transmitted to the joining site by the floor controller. For establishment of coteries, new instances of floors are created for conferees and floor control is applied in this new image of the shared workspace. Recursive subgrouping needs to be reflected in a hierarchy of floors and their inheritance scheme. To find the current floor holder in order to claim the floor, two solutions are possible: either broadcasting to all sites and requesting this floor, or checking directly for its current holder via multicast and asking him to relinquish the floor.
- *Termination and exceptions:* Special cases of temporary and final participant withdrawal must also be handled by floors. For withdrawal, the underlying protocol must notify remote sites of an collaborator's idle state. Through an entry in the floor state table this collaborator is marked or deleted. Imagine that one site owns and controls specific media floors. If another site holds such a floor and crashes, graceful degradation is guaranteed by letting the *orphaned floors* migrate back to the owning site. If the owning site crashes, the floors are revoked from holders and data are withdrawn from the conference, since the crashed site shared its data and no other user can access those data anymore. Migration of the floor controllership to the site of the current holder or an election of a new controller could ensure that a resource can persist as being shared, if its controller site died, but previously declared that resource as permanently public.

We have developed a floor control protocol, called `FACE` (Floor Assignment for Collaborative Environments), which is based on a general taxonomy of floor control properties. It will be used for a *distance-learning* environment within an ATM-testbed to facilitate remote teaching between separate campus units.

5 AN API FOR FLOOR CONTROL

In order to enable an integrated floor attribution service, each application needs a well-defined interface to the floor agent. An application programmer's interface (API) is suggested for manipulating floor states by means of the state table with standardized calls in order to "collaboratize" applications:

- `checkfloor()` returns the Ids of the floor owner and the current floor holder for a given object and whether the respective floor is idle,
- `createfloor()` creates a new floor for a designated resource by making an entry in the floor state table,
- `expandfloor()` sends the current state table to invited or joining users,
- `grabfloor()` is issued, when the floor of a resource is claimed,
- `grantfloor()` attributes a floor of a medium or resource to a specific user,
- `lockfloor()` locks a shared resource,
- `releasefloor()` frees a floor. If nobody requests the floor for that particular shared object, it remains with the floor holder until requested otherwise,
- `revokefloor()` is used by the floorowner to revoke active floors held by others before withdrawing from a session.

Synchronization of floor state tables at all sites must be ensured, but only those sites actually involved in current floor manipulation actions need to update their state tables, based on the suggested "lazy consistency" scheme, i.e., illegal calls on floors have no effect.

6 RELATED WORK

The necessity for floor control in CSCW and CMC systems has been recognized since approximately a decade and is rooted in psychological research on turn-taking behavior in conversations.[23,33,42,44] Experiments to examine the relevance of turn-taking behavior in CMC, compared to face-to-face meetings, have been conducted to increase the effectiveness of CSCW software.[27] However, compared to the full spectrum of desktop conferencing tools and systems, the number of systems currently featuring some notion of floor control, is still rather small. More elaborate classifications for collaborative software have been suggested.[21,26] However, we focus on a few systems, grouped into two categories, because their floor control related design is particularly interesting.

Conferencing systems to support face-to-face meetings as real-world conferencing testbeds, enriched by teleconferencing tools and media used as "catalysts" for communication with "manual" floor negotiation. Examples are experimental environments like the camera-based `DigitalDesk`,[45] which merges digital and real desk work, the shared drawing tools `Clearboard-1/2`,[18] based on glass-boards as digitizer-screens allowing for local work with awareness of remote gestures and processes, the open shared workspace of the `TeamWorkStation`,[19] merging real desktop activities with computer-represented data via a camera interface and translucent overlay, or the concept of media-monitored meeting rooms in `MediaSpaces`.[3]

Conferencing systems to substitute face-to-face meetings, allowing for entirely computer-based conference conduction in distributed sessions. A simple example are textbased conferencing tools with early concepts of floor control in UNIX, coworking via unicast `write` or broadcast `wall` (granting or denying the floor to anyone with `mesg`), `talk` for two-party chatting (turntaking is symbolized by a cursor, jumping between screen halves designated to parties), `ytalk` for 3-way sessions, and `confer` or `joinconf` for multiparty communication. For `confer`, the initiator is the conference leader and designated users are invited automatically. In order to drop out of a conference, an invited guest needs explicit excuse from every participant, and conference proceedings are logged. The floor is claimed by pressing the Enter-key, notifying other participants of the floor holder by displaying his or her name in brackets on all screens. A user is presumed to have the floor until it is relinquished by entering a blank line. Race conditions are simply resolved by granting the floor to the last person claiming it. More sophisticated CSCW and CMC groupware tools, integrating audio, video, facsimile, phone and other communication technology with networked workstation computing environments, provide a "virtual" conference space with computer-mediated floor control:

A first object-oriented architecture for teleconferencing with floor control was proposed by Aguilar et.al.[1] `CoLab`[41] was one of the first collaborative systems, addressing floor control as a conflict resolution strategy based on a dynamic voting scheme. Each site stores data replicated and changes are made on shared parts by unsynchronized broadcasts which are coordinated verbally by the users. The floor is symbolized by a busy signal, graphically warning about editing-conflicts on shared files. To solve the problem of inherent delay between propagation and reception of a warn signal, timestamps or two-phase file locking were considered, specifically a dependency-detection model based on comparison of old and new timestamps and a roving-locks model to create a working set of locks on shared data.

In the collaborative editing and real-time calendar systems, `MPCAL` and `RTCAL`,[38,14] automatic and manual floor-passing are distinguished and a reservation-based floor control scheme has been realized for exclusive updating of shared data. The importance of a smooth conference phasing out is stressed, because typically not all participants leave a conference at the same time. `V` is an integrated multimedia conferencing environment, based on a message-based replicated architecture to minimize network traffic.[22] A conference front-end (user interface and invocation of shared applications), conference agent (mediating I/O between shared applications) and a conference manager (floor control and other synchronization) are introduced as three process types. The system lacks support of voice and long-haul networks and its floor control mechanism can lead to visual inconsistencies between connected sites.

An *activity-sensing* floor acquisition strategy for local area networks has been proposed by Garcia-Luna et.al.,[13] where sites backoff from claiming the floor, when they perceive remote activity. `MMConf`[5] is an architecture for shared real-time conferencing, favoring a centralized server scheme over a replicated approach. Telepointers connecting simultaneous remote activities are managed via floors, one per conversation. Each floor consists of a token with a sequence count to preserve ordering. Each application has one floor manager, communicating with other managers about floor passing. The protocol is unsafe, because applications can refuse to relinquish the floor, or the floor can be in transit, not held by any manager, forcing re-transmissions of a request. If the apparent floor holder's site becomes inaccessible, the least-recently created remaining manager regenerates the floor token based on an out-of-date record.

A framework for shared multimedia workspaces is realized in the system `JVTOS` (Joint Viewing and Tele-Operation Service),[6] integrating session control and a fixed set of floor passing mechanisms implementing different floor policies based on telepointers. A distributed activity-sensing floor control algorithm is realized in `CECED`,[4] based on a pseudo X-server that multiplexes data from tapped links on multicast links to selected sites. `Nebula` is a fault-tolerant conferencing system with three different modes of floor passing within public windows, updating changes within shared data through transmission of update information.[30]

Many dedicated *shareware* applications continuously enter the market,[26] for example shared spreadsheets, joint decision support systems, collaborative CASE tools, interactive hypermedia browsers, shared editors, sketchpads etc. Floor control is provided for example in `Diamond` (replicated shared-view system and basis for `MMConf`), `JointX` (application sharing where floor granting is performed by a moderator via a control panel), `MarkUp` (co-authoring/review system, where collaborative changes to a document are merged after modification – every collaborator has a floor and efforts are integrated a posteriori), `Share` (screen sharing with different floor control modes), `Shdr` (shared drawing with a chalk-passing mechanism for floor-migration), `Sketchpad` (multiuser sketchpad with separate labeled pointers per user), `Talkshow` (multiuser whiteboard with differently colored pens), `XT-confer` (groupware-toolkit with "open" (free) or "closed" (claimed) floors and automatic selective sharing for different media), and `YarnDemo` (chair-guided conferencing, where conferees compete for the floor after each meeting remark).

Multicasting for Multiparty Interactive Multimedia (MIM) applications,[43] e.g., broadcasting services, the "virtual cafe", or distributed computing allows for interaction between participants characterized in dimensions of interaction (static or dynamic group), accessibility (controlled or uncontrolled), and event scheduling (planned or unplanned). It shows that floor control is particularly needed for larger dynamic virtual assemblies. Collaborative visualization systems with partial floor control are `Shastra` for medical imaging,[2] `LinkWinds` for geosciences,[20] `CSpray` for marine and geosciences,[35] and `Highend` for aerodynamics.[34] A conceptual integration of floor control within intelligent agent architectures has been proposed.[9]

Common to the design of these applications are shared public cursors or other widgets for joint visual manipulation of graphically displayed data. Public windows allow for displaying current data of a collaborator, while keeping non-sharable local actions and data invisible within private windows. Floor control is mostly featured only as very basic service.

7 CONCLUSION AND FUTURE WORK

Designing tools for CMC, based on real-world interaction patterns, has to obey certain qualitative and quantitative restrictions to allow for online synergy between users. Since conversations between humans rely to a big extent on nonverbal cues and "social protocols" that cannot be fully conveyed and perceived in the same subtlety within electronic conferencing, session control and floor control need to mediate between users to account for "computer group dynamics".

Limited image resolution, transmission delays, speaker identification and engagement problems occur typically in tele-networking. Cognitive overload by "packed screens" is possible, since all communication is primarily visual and bundled via the workstation. Gestures account for 35% of all interactions in order to enact ideas, signal turn-taking, or reference to objects.[15] Transferring such spatial aspects of face-to-face meetings into the twodimensional desktop metaphor and interactional "bottlenecks" of network links requires hence substitutional tools like a variety of cursors for pointing and gesturing. Even though visual cues proved to be essential in turn-taking, studies on video[17,32] elicited the limits of this channel as interactional vehicle.

Several aspects affecting the design of floor control services in networked multimedia systems have been addressed. On the system's level, an end-to-end service has to provide reliable and efficient multicast routing with fault-tolerance; the application layer above has to provide leveled, fine grained sharing, synchronization and consistency, correct and fair floor assignment, adaptability, and graceful degradation. The user interface needs to accommodate different user roles and reflect the organizational structure and quality of face-to-face meetings as close as possible.

An extension of the local *look and feel* to sharing of remote collaborators' look and feel is needed. Through the user interface, a collaborator can control whether a particular sharable application is working privately or publicly. In the latter case the amount of sharing needs to be adjustable. Also, the *reciprocity* rule[12] needs to be observed – if a collaborator's activities can be perceived remotely, the remote site's activities need to be also transparent locally or at least there is notification about the peers.

Future floor control schemes could incorporate models on discourse structure in speech recognition, in order to capture the *illocutionary force* of the next utterance[28] or prosody for syntactic segmentation and dis-ambiguation, predicting possible turn-taking and transfer of floors. Such analysis can be based also on rhetorical pauses or interrogatives. Instead of making floor assignment mouse-based, it could also be *voice-activated.* A non-intrusive floor control scheme has to maintain the domain information of real-world multiparty collaborations to present a natural feel to each collaborator.[31] Because of the diversity of applications and collaboration parameters a floor control protocol must be *resource-adaptive.* Future GUIs need to provide a *panoramic view* on collaborative actions and involved media to yield awareness for the workgroup and peripheral cues and events.[8,7]

Dynamic sharing of online work is a new paradigm, whose consequences for communication and data processing will show in the next decade, especially with the rise of ubiquitous computing. Software will increasingly offer networked collaborative modes by default. A specification of floor control must hence meet both the system's need for sound integration into a general framework for collaboration and the user's need for transparency and awareness. The concept of floor control is not only applicable to user-related services, as discussed in here, but also for instance at the transport level.

ACKNOWLEDGEMENTS

This work was supported in part by the Office of Naval Research (ONR) under contract N-00014-92-J-1807.

8 REFERENCES

[1] L. Aguilar, J.J. Garcia-Luna-Aceves, D. Moran, E.J. Craighill, and R. Brungardt. Architecture for a multimedia teleconferencing system. In *Proc. Sigcomm'86*, pages 126–136. ACM, August 1986.

[2] V. Anupam, C. Bajaj, D. Schikore, and M. Schikore. Distributed and collaborative visualization. *Computer*, 27(7):37–43, July 1994.

[3] S. A. Bly, S. R. Harrison, and S. Irwin. Media Spaces: Bringing people together in a video, audio and computing environment. *Communications of the ACM - Special Issue on Multimedia in the Workplace*, 36(1):28–47, January 1993.

[4] E. Craighill, R. Lang, M. Fong, and K. Skinner. CECED: A system for informal multimedia collaboration. In *Proc. ACM Multimedia*, Anaheim, CA, August 1993.

[5] T. Crowley, P. Milazzo, E. Baker, H. Forsdick, and R. Tomlinson. MMConf: An infrastructure for building shared multimedia applications. In *Proc. CSCW'90*, pages 637–650, Los Angeles, CA, October 1990. ACM Press, New York, NY.

[6] G. Dermler, T. Gutekunst, B. Plattner, and E. Ostrowski et. al. Constructing a distributed multimedia joint viewing and tele-operation service for heterogeneous workstation environments. In *Proc. Fourth Workshop on Future Trends of Distributed Computing Systems*, pages 8–15, Lisbon, Portugal, September 1993. IEEE.

[7] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proc. CSCW'92*, pages 107–114, November 1992.

[8] P. Dourish and S. Bly. Portholes: Supporting awareness in a distributed work group. In *Proc. ACM CHI'92*, pages 541–7, Monterey, CA, May 1992.

[9] E.A. Edmonds, L. Candy, R. Jones, and B. Soufi. Support for collaborative design: Agents and emergence. *Communications of the ACM*, 37(7):41–47, July 1994.

[10] W.K. Edwards. Session management for collaborative applications. Technical report, Graphics, Visualization & Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, 1994.

[11] C.A. Ellis, S.J. Gibbs, and G.L. Rein. Groupware - some issues and experiences. *Communications of the ACM*, 34(1):38–58, January 1991.

[12] R. S. Fish, R. E. Kraut, R.W. Root, and R.E. Rice. Video as a technology for informal communication. *Communications of the ACM - Special Issues on Multimedia in the Workplace*, 36(1):48–61, January 1993.

[13] J.J. Garcia-Luna, E. Craighill, and R. Lang. Floor management and control for multimedia conferencing. In *Proc. IEEE Multimedia '89, 2nd COMSOC Int. Multimedia Communications Workshop*, Ottawa, Canada, April 1989.

[14] I. Greif and S. Sarin. Data sharing in group work. In *Computer Supported Cooperative Work: A Book of Readings*, pages 477–508. Morgan-Kaufman, 1988.

[15] S. Hayne, M. Pendergast, and S. Greenberg. Implementing gesturing with cursors in group support systems. *Journal of Management Information Systems*, 10(3):43–61, Winter 1993-94.

[16] E. A. Isaacs and J. C. Tang. What video can and can't do for collaboration: A case study. *Internal Report, SunSoft Inc.*, 1993.

[17] E. A. Isaacs and J. C. Tang. Why do users like video? Studies of multimedia-supported collaboration. *Computer Supported Cooperative Work (CSCW)*, 1(3):163–196, 1993.

[18] H. Ishii, M. Kobayashi, and J. Grudin. Integration of inter-personal space and shared workspace: ClearBoard design and experiments. In *CSCW 92 Proceedings*, pages 33–42, November 1992.

[19] H. Ishii and N. Miyake. Toward an open shared workspace: Computer and video fusion approach of TeamWorkStation. *Communications of the ACM - Special Issue on Collaborative Computing*, 34, No.12:36–50, December 1991.

[20] A.S. Jacobson, A.L. Berkin, and M.N. Orton. LinkWinds: Interactive scientfic data analysis and visualization. *Communications of the ACM*, pages 43 – 52, April 1994.

[21] N. Kamel. An integrated approach to shared synchronous groupware workspaces. In *Proc. 4th Workshop on Future Trends of Distributed Computing Systems*, pages 157–163, Los Alamitos, CA, September 1993. IEEE.

[22] K.A. Lantz. An experiment in integrated multimedia conferencing. In *Computer Supported Cooperative Work: A Book of Readings*, pages 533–556. Morgan-Kaufman, 1988.

[23] J. Larrue and A. Trognon. Organization of turn-taking and mechanisms for turn-taking repairs in a chaired meeting. *Journal of Pragmatics*, 19(2):177–196, Feb. 1993.

[24] J.C. Lauwers and K.L. Lantz. Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems. In *Proc. CHI'90*, pages 663–671, April 1990.

[25] G.H. Lerner. Notes on overlap management in conversations: The case of delayed completion. *Western Journal of Speech Communication*, 53(2):167–177, Spring 1989.

[26] P.S. Malm. The unOfficial Yellow Pages of CSCW - groupware, prototypes and projects. Technical report, University of Tromsø, Norway, January 1994. URL *http://www11.informatik.tu-muenchen.de/cscw/yp/*.

[27] A. McKinlay, R. Procter, O. Masting, and R. Woodburn et. al. Studies of turn-taking in computer-mediated communications. *Interacting with Computers*, 6(2):151–171, June 1994.

[28] M. Nagata and T. Morimoto. An information-theoretic model of discourse for next utterance prediction. *Transactions of the Information Processing Society of Japan*, 35(6):1050–1061, June 1994.

[29] K. Narayanaswamy and N. Goldman. "lazy" consistency: A basis for cooperative software development. In *Proc. CSCW'92*, pages 257–264, November 1992.

[30] J.M. Ng, H.H.S. Ip, and P.H.H. Tsang et. al. A distributed multimedia conferencing system. In *Proceedings TENCON'93*, pages 57–60, New York, NY, 1993. IEEE.

[31] D.G. Novick and J. Walpole. Enhancing the efficiency of multiparty interaction through computer mediation. *Interacting with computers*, 2(2):227–246, Aug. 1990.

[32] B. O'Conaill, S. Whittaker, and S. Wilbur. Conversation over video conferences: An evaluation of the spoken aspects of video-mediated communication. *Human-Computer Interaction*, 8(4):389–428, 1993.

[33] D.C. O'Connell, S. Kowal, and E. Kaltenbacher. Turn-taking: A critical analysis of the research tradition. *Journal of Psycholinguistic Research*, 19(6):345–373, Nov. 1990.

[34] H.-G. Pagendarm and B. Walter. A prototype of a cooperative visualization workplace for the aerodynamicist. In *Proc. of the Eurographics'93*, volume 12, No. 3, pages 485–508, 1993.

[35] A. Pang, C. Wittenbrink, and T. Goodman. CSpray: A collaborative scientific visualization application. In *Proc. Multimedia and Networking '95*, San Jose, CA, February 1995. IS&T SPIE.

[36] A. Prakash and M. J. Knister. Undoing actions in collaborative work. In *Proc. CSCW'92*, pages 273–280, November 1992.

[37] 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, April 1991.

[38] S. Sarin and I. Greif. Computer based real-time conferencing systems. In *Computer Supported Cooperative Work: A Book of Readings*, pages 397–420. Morgan-Kaufman, 1988.

[39] E. M. Schooler. The impact of scaling on a multimedia connection architecture. *ACM Multimedia Systems*, 1:2–9, 1993.

[40] H. Shen and P. Dewan. Access control for collaborative environments. In *Proc. CSCW'92*, pages 51–58. ACM, November 1992.

[41] M. A. Stefik, G. Foster, D. Brobrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. In *Computer-Supported Cooperative Work: A Book of Readings*, pages 335–366. Morgan-Kaufman, 1988.

[42] J. Stephens and G. Beattie. Turn-taking on the telephone: textual features which distinguish turn-final and turn-medial utterances. *Journal of Language and Social Psychology*, 5(3):211–222, 1986.

[43] C. Szyperski and G. Ventre. Efficient multicasting for interactive multimedia applications. Technical Report TR-93-017, International Computer Science Institute, Berkeley, March 1993.

[44] M.B. Walker. Smooth transitions in conversational turn-taking: Implications for theory. *Journal of Psychology*, 110(1):31–37, Jan. 1982.

[45] P. Wellner. Interactive with paper on the DigitalDesk. *Communications of the ACM - Special Issue on Computer Augmented Environments*, 36, No. 7:86–97, July 1993.