FLOOR CONTROL FOR ACTIVITY COORDINATION IN NETWORKED MULTIMEDIA APPLICATIONS*

H.-Peter Dommel • *J.J. Garcia-Luna-Aceves*

peter@cse.ucsc.edu • jj@cse.ucsc.edu

Baskin Center for Computer Engineering & Information Sciences

University of California, Santa Cruz, CA 95064

Abstract – Collaboration in networked multimedia applications requires means to coordinate the activities of a dynamically aggregating set of distributed users, working with various multimedia data on heterogeneous platforms. A floor denotes a control right over a shared resource within a collaborative workspace. Floor control, similar to concurrency control for databases, is gradually being integrated into shared applications to orchestrate the access and dynamic process of joint work on shared data, supporting or substituting a human conference chair.

This paper presents a comprehensive view on floor control, analyzing requirements for protocols with respect to the variety of shared tools, describing an architecture to meet these requirements, and finally placing our work in the context of previous efforts.

Keywords – Floor control, collaborative multimedia computing, Computer-Supported Cooperative Work (CSCW).

1. Introduction

For multimedia applications, a gradual shift from standalone to networked environments can be observed. Internet applications demonstrate a popular demand for online sharing of information. However, data sharing occurs mostly on static results from finalized work efforts. The new trend of dynamic collaboration in on-going work by means of a set of integrated applications among members of a workgroup allows for an extension of the prevalent WYSIWIS-paradigm (*What You See Is What I See*) to a selective WYSIWISH-paradigm (*What You See Is What I SHare*). The scope of local platforms and applications is enhanced to local-area or wide-area collaborative online meetings and man-machine interactivity is extended to a man-machine-man *collaborability* as a new dimension on top of compatibility, interoperability, and portability.

Simple models of *groupware* have been implemented a decade ago. New shared environments with increasing functionality and complexity allow for multipoint, multiparty, multichannel, and multimedia communication. For such teleconferencing applications, new protocols for managing the formation of online meetings, called *sessions*, and for handling the variety of multimedia streams in collaborative work are needed.

In comparison to the quality of face-to-face meetings, computer-mediated remote interaction has several drawbacks: there is no contextual view of the meeting scenario, "flat" user interfaces are used for mediation between parties, often reducing the full quality of the presented information, and social conventions conveyed in personal presence via visual cues, deictic and mimic gestures are mostly not applicable.

Especially for large sessions with fluctuating membership, a mechanism has to be introduced to support or replace a chairperson in assigning activity permissions to specific participants within the open shared workspace. Of course, the ultimate test is the acceptance by users, making the Quality-of-Service (QoS) of such a mechanism a function of system and usability parameters in order to achieve *telepresence*.

Floor control, targeted at the application-level, extends the notion of database *concurrency control* to online shared multimedia objects, but relates to distributed *access control* [25] for files and *admission control* for transmission channels as well. Floor control in CSCW is a metaphor for "assigning the floor to a speaker", which is applicable not only to voice-channels, but more generally to any kind of sharable resource within conferencing and collaboration environments. Conceptually it is a dynamic counterpart to *version control*, as applied in software engineering, and analogies can also be found in many real-world problems requiring mutual exclusion, cf. *ground control* in traffic studies, or *semaphors* and *monitors* in parallel processing.

A *floor* is an individual temporary access or manipulation permission for a specific shared resource, e.g., a telepointer or voice-channel, allowing for concurrent and conflict-free resource access by several conferees. Through floors, race conditions for resources in shared

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

work can be mitigated or, ideally, prevented *a priori*. We discuss important requirements for floor control protocols and a basic architecture to allow for adaptive control of sharing any kind of multimedia resource within distributed collaborative groups.

2. Requirements for Floor Control

For the design of floor control services[1] the systems' as well as the users' perspective are equally important [6], since floor control is an user-endorsed system aid. The following service criteria are crucial:

- *distributed server control* for individual applications and tracking of floors for the sake of *scalability* [24] in large workgroups, *resilience* in case of drop-outs and site-crashes, *efficiency* with respect to *multipoint* control message traffic and *responsiveness* in floor attribution,
- *correctness* with respect to *liveness*, i.e., deadlock-freedom in floor-assignment,
- *fairness*, designating a reliable and balanced floor policy for all users, although preemption must be possible to override automatic by manual floor assignment,
- *adaptability* with respect to heterogeneous platforms and varying user preferences as well as *extensibility* for new types of shared resources,
- *security* despite *floor transparency*, i.e., specific conferees can intelligibly access any otherwise secured resource in a collaborative domain,
- *usability* for the sake of *acceptance* and *seamlessness* [26] of the intra- and inter-application integration of different media with semantic and temporal synchronization of collaborating sites.

A floor control mechanism has to accommodate a variety of parameters characterizing a teleconferencing or collaboration scenario facilitated by some *session control service* [9]:

Session parameters entail the number of collaborators, their aggregation into (sub)groups, and roles (chair, floor holder etc.) determining their capabilities. Also, their interconnectivity (1-1, m-n), sharing distribution range (local, wide-area, global) and link-types ((un)restricted, bi- or unidirectional) are crucial. For applications to scale beyond a few participants, all communication must be multicast. Resources vary with the applications involved, encompassing telepointers, customized widgets, files, events, windows and views, video and audio channels, still and motion image sequences, virtual spaces, and further software or hardware components. Floors are characterized in configuration (preassigned statically or relocated dynamically), authorization (primary or feedback rights), instantiation (single, or for media like voice with possibly several concurrent speakers, multiple), policy (automatic or chair-guided), and longevity (usage bounded by time, event-occurrence, resource-demand, etc.). These parameters together configure single floors in a causal chain and determine control of the sharing process.

As of now there is no comprehensive notion of QoS in multimedia environments, comprising "hard" network and system parameters like transmission delay bounds as well as "soft" user-related parameters such as turn-taking behavior. The floor control protocol has to entail QoS guarantees at the endsystem level [11] based on the QoS of lower levels, e.g., switching capacity, or buffer space in ATM cross connects. TCP is insufficient, in that its socket abstraction does neither provide resource allocation obeying QoS parameters, nor real-time delivery guarantees or multiparty communication. Work on multimedia real-time protocols is meant to solve these shortcomings.

A floor control protocol has to ensure that conflicts on resources are avoided via an assignment policy that is viable for all users, preventing inconsistencies in the shared work process through mutual exclusion. However, since manual floor control can interfere and inconsistencies in shared data states are possible, a synchronization or regeneration mechanism for making remote sites consistent is needed as well. Negotiation of a floor for a shared object is not only a matter of its availability, but also of the prospect to have sufficient resources available to satisfy the activity. Furthermore, some media like voice and video streams require strict real-time delivery and synchronization, but tolerate some lossiness, whereas textual or graphical objects, e.g., in a collaborative whiteboard, are lossless, but can incur some delay. Floor control has to adapt to these timeliness requirements. We present now briefly a principal architecture to attain floor control of shared multimedia objects and activities, and outline a protocol observing the above specifications.

3. Floor Control Protocol Realization

The requirements motivate an implementation where a floor daemon on each node in a collaboration graph controls local floor assignment of locally owned, but shared resources, synchronizing with remote nodes. It

[1] Although we focus here on floor control as an application-level concept, it is also applicable to end-to-end services.

interfaces with a session control protocol, which orchestrates sites to reach consensus on group membership [21] and channel establishment. An object-oriented model fosters distinction between private and public data, as well as object linking and inheritance in hierarchical session control [22]. Floors can not only attach to media, but also to sessions, permitting or refusing to join certain meetings. A principal protocol stack is depicted in Figure 1. Session control focusses on general facilitation of online meetings, whereas the floor control addresses aspects of work coordination, authorization and resource sharing.

Layer	Function
application, X	resource coupling, sharing interface
floor control	authorization, access, activity coordination
session control	orchestration, authentication, synchronization
network	reliable end-to-end-services

Figure 1: Basic floor control architecture.

We distinguish between the contributor or *floor controller* of a specific resource, the person with temporary activity rights on that resource, called the *floor holder*, and regular session attendees as collaboration bystanders and tentative floor holders. Floor control principles are based on standard concurrency control like two-phase locking, but must accommodate the interactive nature of collaboration between users. Resource dependency detection, resource reservation, and dynamic voting on a floor holder are currently employed techniques, based on active token passing or passive resource-activity sensing to achieve mutual exclusion between critical work on shared data. Furthermore, different *policies*, i.e., scheduling and queueing techniques for floors requests, need to be offered within the same floor control *mechanism* on all sites to allow for adaptation to different resources. Examples are chair-guidance, round-robin, demand-intensity, first-come-first-served, and least-recently-served.

The resource-adaptive protocol FACE *(Floor Assignment in Collaborative Environments)* [7] operates on the above premises. FACE features contention avoidance without predefined token scheduling, and allows for automatic or chair-guided conference facilitation. It features 4 floor states, designating local or remote floor attribution, and it is adaptive by using resource type descriptors incorporated in message packets to check for usage authorization of different kinds of media. A basic premise is that no failures occur on links. To ensure fault tolerance, a fifth protocol state characterizes exceptions like site-crashes and link-failures, inciting a distributed election algorithm to regenerate a stable scenario, if necessary by determining new controllers and holders for orphaned floors. The control packets sent between sites contain identifiers on the session, host, group and subgroup, the collaboratee and role within the session, the application with adjunct shared resources, and finally, the specific floor. Selection of a floor holder is multicast to involved sites based on the request label and the used assignment policy.

Since a large set of conferencing parameters has to be tracked on every site for all users and resources, each workstation must have the computing resources to deal with the protocol and interoperability overhead implied by the usage of heterogeneous platforms. For standardization and extensibility, an application-programmers-interface (API), as outlined in Figure 2, is needed.

```
checkfloor()    grantfloor()
createfloor()   lockfloor()
expandfloor()   relinquishfloor()
shrinkfloor()   revokefloor()
claimfloor()    killfloor()
```

Figure 2: Set of basic calls in floor control API.

User-interfaces designed for standalone work or mere replication of views have to be redesigned for true information sharing. One approach, based on a modification of X, is to *drag-and-share* in a "virtual shared desktop". With this paradigm a resource becomes public across connected sites, if it is declared as shared by dragging it into a symbolic "sharing-pool window", making it visible and ready for coupling to involved sites. For every user, a pull-down list of his momentary public resources must be available. A more sophisticated representation can be based on a semantic net with zoom-in capabilities [23], reflecting the hierarchical nature of the session model and allowing for entering and leaving of specific sessions and levels via the GUI (graphical user-interface). Floor states can be depicted by visual or auditory cues, e.g., coloring a shared objects in red depicts a used floor and locked resource. Floor policies, usage allowance time etc., must be adjustable in menues and presetable in configuration files. To allow for replay of tool usage and monitoring, a logging mechanism is useful. Automatic floor migration can be triggered based on time or events, or via periphery, e.g. mouse-buttons or data-glove gestures. Overall, user-acceptance can be fostered via non-intrusiveness of floor assignment, accessibility and transparency in the GUI.

4. Related Work

Roots of floor control research in the context of CSCW and Computer-Mediated-Communication (CMC) can be found in cognitive research on *turn-taking* behavior in conversations in order to increase the quasi-face-to-face effectiveness of CMC [16, 19]. Looking at the variety of groupware [18], existing systems can be coarsely categorized in two groups:

1. systems *supporting face-to-face meetings in real-world conferencing*, e.g., via a camera-based DigitalDesk [27], Clearboards [13] as digitizer-screens allowing for local work with awareness of remote gestures and processes, TeamWorkStations [14], merging real desktop activities with computer-represented data via a camera interface and translucent overlay, or media-monitored meeting rooms in MediaSpaces [2]. Such testbeds have served as "catalysts" for studies in remote communication with "manual" floor negotiation.

2. systems *"virtualizing" and substituting face-to-face meetings*, allowing for entirely computer-based conference conduction in distributed sessions. Within this paradigm, we can identify three major categories, mentioning a few systems among many existing ones, which were significantly innovative with respect to floor control:

- *Collaboration-unaware* systems focus on window synchronization, making sharing an interface-oriented add-on to the application with floor control as a "spy-mechanism" to trace and filter collaborative requests:

CoLab [26] was one of the first collaborative systems, addressing floor control as a conflict resolution strategy based on a dynamic voting scheme. Sharing is based on verbally coordinated and unsynchronized broadcasts and the floor, symbolized by a busy signal, warns graphically about editing-conflicts. Timestamps and two-phase file locking were employed. Automatic reservation-based and manual floor-passing are distinguished for MPCAL and RTCAL, collaborative editing and real-time calendar systems [12]. The VConf system [15] utilizes floor control via a "conference manager" interfacing with a user front-end and an agent mediating the I/O between shared data. A centralized real-time conferencing approach is favored in MMConf [4], where floor-controlled telepointers connect simultaneous remote activities. Floors are assigned in sequence via token, and each site has one floor manager, communicating with other managers about floor passing. The employed protocol is unsafe, since 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. JVTOS [5] integrates session control with a fixed set of floor passing policies on telepointers. A distributed activity-sensing floor control algorithm was realized in CECED [3], based on a pseudo X-server that multiplexes data from tapped multicast links to selected sites.

- *Collaboration-aware* systems feature inherent support of resource-linking and collaborative activities: 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" or "closed" floors and automatic selective sharing for different media), and YarnDemo (chair-guided conferencing with user-competition for the floor after each contribution). Public-domain conferencing for the MBone (virtual internet Multicast IP Backbone) [10] includes the video tool vic, the whiteboard wb, and the visual audio tool vat, which supports voice-activated floor switching. Some coherency for these independent and experimental tools is provided via integration into the session control directory sd.

- *Collaboration-transparent* [17] systems are dedicated applications using generic conferencing tools for text, video and audio conferencing as enrichments to their inherent collaboration architecture, making them a hybrid of the first two categories. Examples are collaborative visualization systems like Shastra for medical imaging [1], and CSpray for marine sciences [20]. Both systems supply a notion of floor control within asymmetric workspaces, with the latter system serving as our testbed for floor control issues. Recently, the conceptual integration of floor control within intelligent-agent architectures has been proposed [8].

Drawbacks of current systems are that floor control is still in its infantile stage. Long-haul networks or large-scale conferencing are not supported, many performance problems can be observed with higher volume data collaboration, data inconsistencies across coupled sites can occur, and sharing focuses only on few media with simplistic floor policies.

5. Conclusion and Perspective

Existing systems show the many faces of floor control. There is a lack of software designed to coordinate and control various interrelated media and research on floor control is intended to alleviate this. "Every access to every (shared) object should be checked for current authority" is the axiom of *total mediation* [25], however, only few applications in the current spectrum of CSCW software feature a notion of floor control for any type of shared object. Future research needs to integrate results from both the systems level as well as human factors, looking at a message ordering semantics for multicasting as well as at user-modeling and interface issues. Graphical user interfaces will have to be extended towards shared multimedia presentation capabilities and incorporate a notion of a "panoramic view" of conference surroundings to approximate face-to-face meeting quality. Not only will future multiprotocol suites for collaboration have to be self-adapting to the heterogeneity of platforms and software environments, but display degrees of "learnability" towards the users served and the services to be provided.

Our approach is not intended as a "panacea" for conferencing environments and any kind of media, but as another integrating step towards a more flexible, comprehensive and rich notion of collaboration, where groupwork is facilitated and secured. Currently we work on implementing an API to realize an increasing subset of a full-fledged floor control service within the BayLink ATM-testbed, supporting collaboration between marine scientists, providing information service to schools and museum visitors, and experimenting with distance learning between our university and its remote extension facility.

In the long run, floor control, as an essential concept of collaborability, will be an integral part of collaborative software. More challenges wait in the form of ubiquitous computing where users will join sessions via faulty links from wireless hand-held devices or mobile video terminals.

References

[1] V. Anupam, C. Bajaj, D. Schikore, and M. Schikore. Distributed and collaborative visualization. *Computer*, pages 37–43, July 1994.

[2] S. A. Bly, S. R. Harrison, and S. Irwin. Media Spaces: Bringing people together in a video, audio and computing environment. *CACM – Special Issue on Multimedia in the Workplace*, 36(No. 1):28–47, January 1993.

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

[4] 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, October 1990.

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

[6] H.-P. Dommel and J.J. Garcia-Luna-Aceves. Design issues for floor control protocols. In *Proc. Multimedia and Networking '95*, San Jose, CA, February 1995. IS&T SPIE.

[7] H.-P. Dommel and J.J. Garcia-Luna-Aceves. A floor control protocol for networked multimedia applications. Univ. of Santa Cruz, Baskin Center for Computer Sciences and Engineering, 1995.

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

[9] W.K. Edwards. Session management for collaborative applications. Graphics, Visualization & Usability Center, College of Computing, Georgiatech, Atlanta, Internet Draft, 1994.

[10] V. Kumar et.al. The MBone information web homepage. URL *http://www.eit.com/techinfo/mbone/mbone.html*, since 1992.

[11] R. Gopalakrishnan and G.M. Parulkar. Application level protocol implementations to provide quality-of-service guarantees at endsystems. Department of Computer Science, Washington University in St. Louis, Internet Draft, December 1994.

[12] 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.

[13] 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.

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

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

[16] 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.

[17] 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.

[18] P.S. Malm. The unOfficial Yellow Pages of CSCW – groupware, prototypes and projects. Technical report, University of Tromsø, Norway, January 1994. URL *http://www.tft.tele.no/cscw/*.

[19] 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.

[20] 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.

[21] B. Rajagopalan. Consensus and control in wide-area group communication. Technical report, AT&T Bell Laboratories, Holmdel, NJ 07733-3030, October 1993. Internet Draft.

[22] 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.

[23] E. Rennison. Galaxy of news and information landscapes: Dynamic visualization and access of information in a multidimensional space. Demonstration at ACM Multimedia '94, San Francisco, October 1994.

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

[25] H. Shen. *Access Control for Collaborative Environments*. PhD thesis, Purdue Univ., Lafayette, IN, August 1994.

[26] M. A. Stefik and G. Foster et. al. 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.

[27] P. Wellner. Interactive with paper on the DigitalDesk. *CACM – Special Issue on Computer Augmented Environments*, 36, No. 7:86–97, July 1993.