

14

Narrowcasting in SIP: Articulated Privacy Control

Sabbir Alam, Michael Cohen, and Julián Villegas

University of Aizu

Ashir Ahmed

Kyushu University

CONTENTS

14.1 Introduction	323
14.2 Conferencing	325
14.3 Media Privacy: Narrowcasting Concept	328
14.4 System Design and Implementation	332
14.5 Conclusion and Future Research	339
Acknowledgements	343
References	343

14.1 Introduction

In multimedia conferencing, media streams are exchanged between participants upon session establishment by setting up communication channels within a group. By default, each participant receives a combined stream obtained by mixing other participants' media. Situations arise when one wants to select a subset of the conference participants to whom one's media is sent or from whom streams are received. Media filters are necessary to configure privacy of the participants in the conference. In analogy to broad-, multi-, any-, and swarm-casting, narrowcasting is a technique for limiting and focusing information streams. Narrowcasting systems extend broad- and multicasting systems by allowing media streams to be filtered—for relevancy control, privacy, and user interface optimization. We describe four narrowcasting commands: `mute`, `deafen`, `select`, and `attend`—to provide distributed privacy in SIP-based conferencing.

Extensive Development has been carried out in the area of conference and floor control [1,2]. Conventional features regarding media privacy in conferences are typically limited to scheduling and selecting the speaker. Advanced conferencing features such as adding/deleting participants, changing user agents (UAs) or modes (like switching from a desktop to a mobile phone),

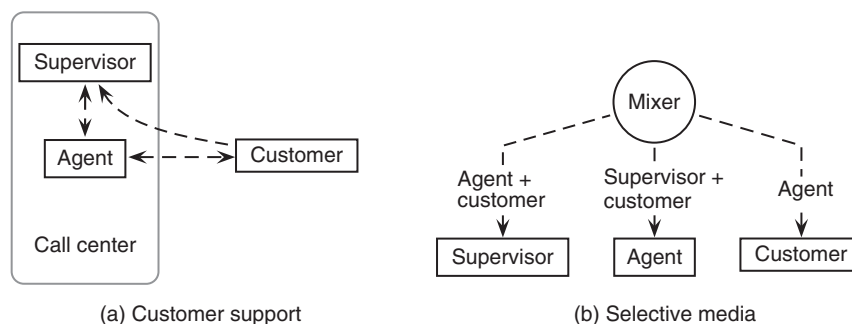
TABLE 14.1
Three different mute operations.

	Self-mute	PBX mute	Narrowcasting mute
Media vector			
Media distribution	$P_1 \leftarrow (P_2 + P_3 + P_4)$ $P_2 \leftarrow (P_3 + P_4)$ $P_3 \leftarrow (P_2 + P_4)$ $P_4 \leftarrow (P_2 + P_3)$		$P_1 \leftarrow (P_3 + P_4)$ $P_2 \leftarrow (P_1 + P_3 + P_4)$ $P_3 \leftarrow (P_1 + P_2 + P_4)$ $P_4 \leftarrow (P_1 + P_2 + P_3)$
Semantics	Self control. P_1 mutes himself by turning off his mic so that no media goes to the media server, or P_1 can send a “self-mute” signal to the application server so that the media server simulates self-censorship.	Control by admin. P_1 is muted by moderator. P_1 ’s media is not mixed in the media server. P_1 is in a listen-only, or “lurker” (stealth) mode.	P2P Control. P_1 mutes P_2 . P_2 may speak to everyone, but P_1 won’t hear his voice.

changing media, authenticating or authorizing participants, granting privileges, controlling presentation of media, sidebars, passive participants, whisper/private messages, audio-only, and lecture mode are described in RFC 4597 [3]. Media privacy features allow participants to control their own information and to distribute their attention, based on secrecy, anonymity, and solitude [4].

Mute is a popular feature for media privacy. It has three different varieties, juxtaposed in Table 14.1: self-mute, PBX-mute, and narrowcasting mute. Self-mute allows a user to withhold his media streams from other participants. In PBX-mute, a moderator disables a participant’s outgoing media to other participants. Narrowcasting mute refers to P2P control with which a participant (controller) can select another participant (controllee) to disallow the controllee’s media towards the controller.

A Call Whisper [5] feature allows a participant to talk privately to one or more participants in a group. This walkie-talkie-like feature creates a one-way voice or video communication channel. The session terminates when the controller releases the push to talk (PTT) button, so such a system is not practical when two-way communication is necessary. Voice Chat [6] allows participants to create one or more private audio conferences: although the communication channel in the private voice chat group provides duplex communication, participants can hear the main conference at low volume. Private conversation [7] offers a private video, voice, and text conversation session inside a main conference. It is similar to a Call Whisper feature, but adds duplex

**FIGURE 14.1**

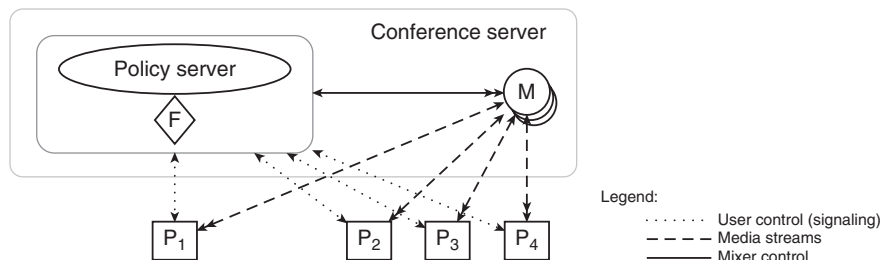
Media privacy: A call center application scenario.

communication capability and text messaging. In a WebEx audio conference, a conference chairperson can selectively disable microphones to allow only certain attendees to speak. An “audio-only” option allows a moderator to revoke and restore speaking privileges to attendees, so that muted attendees can only lurk, listen but not speak. WebEx participants can have a private chat with someone during a meeting. Whisper Coaching (www.audiocodes.com) allows a supervisor to listen to a main conference conversation while talking to a selected set of participants. The privacy control allowed by these applications is rather blunt. In order to better control media privacy, we are exploring the concept and practical applications of narrowcasting [8–10].

A call center scenario provides an example of media privacy: in instances when a first-tier agent cannot answer a customer’s questions, the agent might have a private side-channel communication with a supervisor as back-up for realtime customer support, as shown in Figure 14.1a. Even though the supervisor can overhear the customer, privacy control is invoked so that the supervisor’s media goes only to the agent, not to the customer, as shown in Figure 14.1b. Traditional conferencing systems do not generally provide such features. In this chapter, we describe a mechanism and instance of “Media Server Component Model” architecture for policy-based media mixing with a centralized media mixer, using the standard SIP [11] framework for multimedia conferencing systems. We have defined privacy commands and developed a policy evaluation algorithm: media mixing and delivery factoring policy configured by conference participants.

14.2 Conferencing

A conference server and the participants are two major components of a centralized conference system, as shown in Figure 14.2. A SIP conference server comprises a focus, policy server, and media mixer. The focus handles the

**FIGURE 14.2**

Typical conference architecture: F is the focus, M are media mixers, and P_i are the participants.

conference control—creating, modifying, and terminating conferences. Conference connectivity is managed by the policy server, which can configure the media server. Mixing and distribution of media streams are the main functions of a media mixer, such as a “voice switch” for audio conferencing, which transmits some composite signals to the respective terminals, as suggested by the multiple arrowheads on the (dashed) return vectors. Value-added services—such as monitoring conference status, participant status, and billing—can be implemented inside or outside of this framework.

14.2.1 SIP Conference Model

There are two generic conference models: loosely and tightly coupled. In a loosely coupled model, there is neither a central point of control nor a conference server, whereas in a tightly coupled model, a centralized conference control server manages the conferences. A tightly coupled conferencing model can be further classified into six different types depending on the location of the focus and the mixer, as illustrated in Table 14.2, including the Media Server Component Model used for our proof-of-concept. These models are detailed by J. Rosenberg [12] and Y. Cho et al. [13].

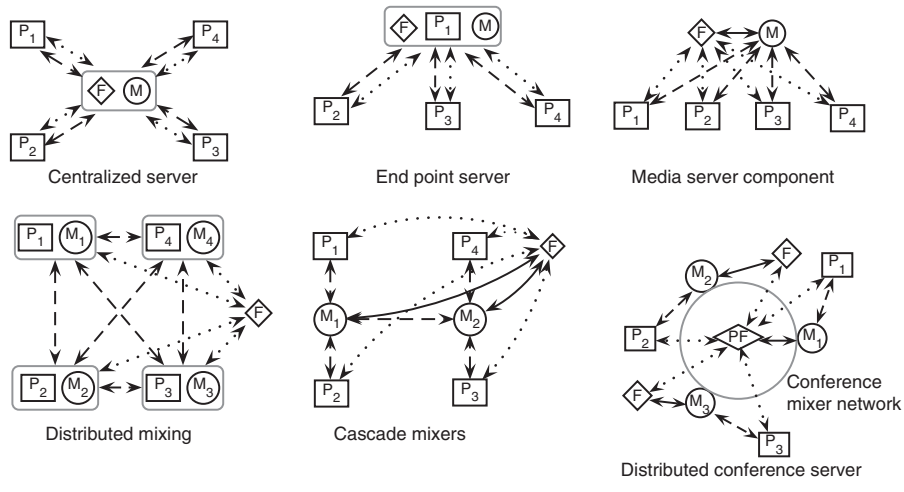
14.2.2 SIP Conference Control

Conference control refers to the ability to manipulate the state of a session. A conference is represented by a unique Uniform Resource Identifier (URI), usually a SIP URI, that identifies the focus of a conference. A conference URI can be e-mailed, sent in an instant message, linked on a web page, or obtained from some other mechanism. Conference control includes three primary functions:

- **Creation:** A participant joins a conference by sending an INVITE request to its focus (“dial-in”) or by the focus sending an INVITE request to the participant (“dial-out”), citing the conference URI.

TABLE 14.2

Conferencing models: “P” indicates participant, “F” indicates focus, “M” indicates media mixer, and (in the last model) ‘PF’ indicates primary focus. Dotted lines indicate signaling, dashed lines indicate media transmission, and solid lines indicate mixer control.



- **Modification:** A participant or focus can modify a session in a conference using a re-INVITE. For instance, when an audio conference extends to video, the focus re-INVITES each participant adding a video media stream. A participant or focus may also put media streams on hold or take them off hold. Narrowcasting commands are applied to a session by selectively enabling the media streams.
- **Termination:** A privileged participant (typically a moderator or conference creator) closes a session by sending a BYE request to the focus. The focus then distributes a BYE request to all other participants in the conference, terminating the session.

14.2.3 Conference Privacy

Privacy has two interpretations. The first association, with sources, is that of avoiding “leaks,” or protecting secrets. But a second interpretation, with sinks, means freedom from disturbance; in the sense of solitude, not being bothered by irrelevance or interruption, as suggested by Figure 14.3. Our distributed interface features narrowcasting operations that manage privacy in both senses, by filtering duplex media flow through an articulated conferencing model that limits and focuses information streams.

**FIGURE 14.3**

Privacy: Freedom from disturbance. (© The New Yorker Collection 1996 Sam Gross from cartoonbank.com. All rights reserved.)

14.3 Media Privacy: Narrowcasting Concept

In traditional conferencing systems, participants have little or no privacy, as their voices are by default shared with all others in a session. Such systems cannot offer participants options for muting and deafening other members. The concept of narrowcasting can be applied to make these kinds of filters available in multimedia conferencing systems. A symmetric model treats media sinks (such as, listeners) as full citizens, peers of the media sources (conversants' voices), and we defined therefore duals of *mute* & *select*: *deafen* & *attend* respectively block a sink or focus on it to the exclusion of others. Figure 14.4 shows a famous carving which informally illustrates multimodal narrowcasting. Three monkeys—Kikazaru (blocked ears), Iwazaru (covered mouth), and Mizaru (with covered eyes)—manifest the notion of



FIGURE 14.4
Media privacy (narrowcasting features).

limiting media vectors. Kikazaru can not hear but can speak and see; Iwazaru can not speak but can see and hear; Mizaru can not see but can hear and speak.

For modern groupware situations like teleconferences, in which everyone can have a presence across the global network, users want to shift and distribute attention (apathy) and accessibility/availability/exposure (privacy), and narrowcasting provides a formalization of such filters. The narrowcasting predicate calculus [14], shown in Figure 14.5, is an appropriate basis for such a permission scheme.

Users are represented by objects in an interface which have attributes corresponding to narrowcasting state. To distinguish between operations reflexive

<p>The general expression of inclusion is:</p> $\text{active (object } x) = \neg \text{exclude } (x) \wedge (\exists y (\text{include } (y) \wedge (\text{self } (y) \Leftrightarrow \text{self } (x))) \Rightarrow \text{include } (x)). \quad (1)$ <p>So, for mute and select (solo), the relation is:</p> $\text{active (source } x) = \neg \text{mute } (x) \wedge (\exists y (\text{select } (y) \wedge (\text{self } (y) \Leftrightarrow \text{self } (x))) \Rightarrow \text{select } (x)), \quad (1a)$ <p>mute explicitly turning off a source, and select disabling the complement of the selection (in the spirit of "anything not mandatory is forbidden"). For deafen and attend, the relation is:</p> $\text{active (sink } x) = \neg \text{deafen } (x) \wedge (\exists y (\text{attend } (y) \wedge (\text{self } (y) \Leftrightarrow \text{self } (x))) \Rightarrow \text{attend } (x)). \quad (1b)$
--

FIGURE 14.5
Formalization of narrowcasting and selection functions in predicate calculus notation, where ‘ \neg ’ means “not,” ‘ \wedge ’ means conjunction (logical “and”), ‘ \exists ’ means “there exists,” ‘ \Rightarrow ’ means “implies,” and ‘ \Leftrightarrow ’ means mutual implication (equivalence).

(invoked on oneself) and transitive (on others), the attributes include a `self` flag, applied to both sources and sinks. The duality between source and sink operations is tight, and the semantics are analogous: an object is inclusively enabled by default unless it is explicitly excluded (with $\overbrace{\text{mute}}^{\text{source}}$ or $\overbrace{\text{deafen}}^{\text{sink}}$), or, peers of the same self/non-self class are explicitly included (with $\overbrace{\text{select}}^{\text{sources}}$ or $\overbrace{\text{attend}}^{\text{sinks}}$) when the respective object is not. Narrowcasting attributes are not mutually exclusive, and the dimensions are orthogonal. Because a source or sink is active by default, invoking `exclude` and `include` operations simultaneously on an object results in its being disabled. For instance, a sink might be first attended, perhaps as a member of some non-singleton subset of a chatspace's sinks, then later deafened, so that both attributes are simultaneously applied. (As audibility is assumed to be a revocable privilege, such a seemingly conflicted attribute state disables the sink, whose receptivity would be restored upon resetting its deafen flag.) Symmetrically, a source might be selected and then muted, akin to inclusion on a "short list" but relegated to back-up.

Our system allows each user to send or receive data streams to/from a specific recipients in a session. For easier understanding, we consider only audio streams in this chapter, but the design applies to other media types. Narrowcasting audio commands are listed and their characteristics arrayed in Table 14.3.

In this section, we formally define four narrowcasting commands. In the following expressions, P_a denotes the actor (controller), P_o the object (controllee), P_i a sender of the media (source), P_j a receiver of the media (sink), for $a, i, j, o \in \{1, \dots, n\}$, where n is the total number of participants.

14.3.1 Mute

The narrowcasting command `mute` blocks media coming from a source. The mute in traditional systems is a self-mute function which allows a user to withhold his/her media from other participants, but the modern `mute` is a control function that can select another participant (or a group of participants) to disallow media towards the controller, still allowing other participants to hear the controllee. The Σ operator composites media from the respective participants.

$$P_j \leftarrow \begin{cases} \sum_{i=1}^n P_i - P_j - P_o & \text{when } \overbrace{P_j = P_a}^{\text{transitive}} \text{ or } \overbrace{P_a = P_o}^{\text{reflexive}}, \\ \sum_{i=1}^n P_i - P_j & \text{otherwise.} \end{cases} \quad (14.1)$$

The example modeled by the matrix in the first column of Table 14.3 illustrates when P_1 mutes another participant P_2 . In this example, $n = 4$, $P_a = P_1$

TABLE 14.3
Narrowcasting commands.

	P_1 mutes P_2	P_1 deafens P_2	P_1 selects P_2	P_1 attends P_2
Media vectors				
Semantics	Block the media stream coming from a source.	Block media streams going to a sink.	Limit projected sound to particular sources.	Limit received sound to particular sinks.
Situation	Participant wants to block media from specific participants.	Participant wants to block media to specific participants.	Participant wants to receive media only from particular participants.	Participant wants to send media to specific participants.
Figurative avatars				
Mobile icons	-△	-△-	+△	+△+
Media distribution	$\begin{bmatrix} \times & 1 & 1 & 1 \\ 0 & \times & 1 & 1 \\ 1 & 1 & \times & 1 \\ 1 & 1 & 1 & \times \end{bmatrix}$	$\begin{bmatrix} \times & 0 & 1 & 1 \\ 1 & \times & 1 & 1 \\ 1 & 1 & \times & 1 \\ 1 & 1 & 1 & \times \end{bmatrix}$	$\begin{bmatrix} \times & 1 & 1 & 1 \\ 1 & \times & 1 & 1 \\ 0 & 1 & \times & 1 \\ 0 & 1 & 1 & \times \end{bmatrix}$	$\begin{bmatrix} \times & 1 & 0 & 0 \\ 1 & \times & 1 & 1 \\ 1 & 1 & \times & 1 \\ 1 & 1 & 1 & \times \end{bmatrix}$

(the controller), and $P_o = P_2$ (the controllee). Due to this operation, P_1 will not receive any media from P_2 . (This is actually a simplification of the evaluation performed by our system, since our model supports multipresence, the designation by a single human user of possibly multiple iconic representatives in an interface. Such complicated subtleties are beyond the scope of this chapter.)

14.3.2 Deafen

Deafen is a sink-related media privacy command that blocks media streams to a selected participant. For example, if Bob (P_1) wants to share his media with everyone in a conference except Alice (P_2), then Alice will not receive any streams from Bob if Bob deafens Alice. (Transposing the participants suggests an equivalent operation via reciprocity of sources and sinks, P_2 mutes P_1 .) The second column of Table 14.3 shows the media relationship among

four participants when `deafen` is invoked.

$$P_j \leftarrow \begin{cases} \sum_{i=1}^n P_i - P_j - P_a & \text{when } \overbrace{P_j = P_o}^{\text{transitive}}, \\ \phi & \text{when } \overbrace{P_a = P_o}^{\text{reflexive}}, \\ \sum_{i=1}^n P_i - P_j & \text{otherwise.} \end{cases} \quad (14.2)$$

Again in this example, $n = 4$, $a = 1$, and $o = 2$.

14.3.3 Select (Solo)

The privacy command `select` limits received media to particular sources. For instance, students might `select` a teacher to avoid distractions. P_1 will receive media only from P_2 if P_1 `selects` P_2 , implicitly muting the complement of the selection. The third column of Table 14.3 shows the media relationships among four participants when `select` is invoked; two vectors are disabled in this case.

$$P_j \leftarrow \begin{cases} P_o & \text{when } P_j = P_a \text{ and } \overbrace{P_a \neq P_o}^{\text{not reflexive}}, \\ \sum_{i=1}^n P_i - P_j & \text{otherwise.} \end{cases} \quad (14.3)$$

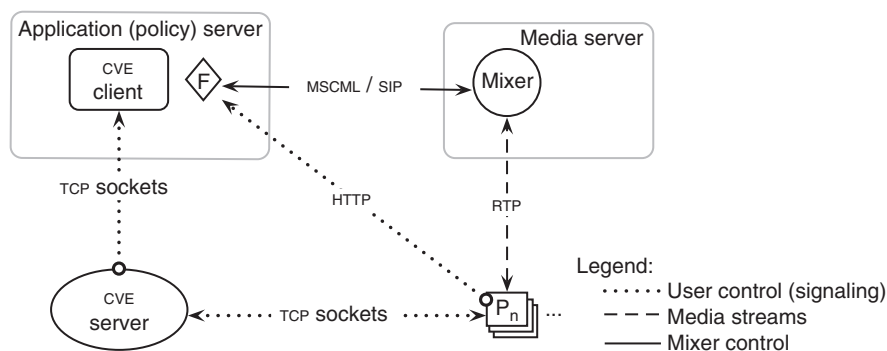
14.3.4 Attend

`Attend` is the other including command for narrowcasting, limiting received media to a particular recipient. If Alice `attends` Bob, only Bob will hear Alice, since other participants are implicitly `deafened`. The rightmost column of Table 14.3 shows the media relationship among four participants when `attend` is invoked; again two media vectors are suppressed.

$$P_j \leftarrow \begin{cases} \sum_{i=1}^n P_i - P_j & \text{when } P_j = P_o \text{ and } \overbrace{P_a \neq P_o}^{\text{not reflexive}}, \\ \sum_{i=1}^n P_i - P_j - P_a & \text{otherwise.} \end{cases} \quad (14.4)$$

14.4 System Design and Implementation

The main required functions for media control are policy configuration, policy evaluation, media mixing, and media distribution. The Media Server Component Model (top right of Table 14.2) selected for our implementation comprises a centralized focus (collocated with the policy server), a centralized mixer,

**FIGURE 14.6**

Media Server Component Model with collaborative virtual environment integration.

and distributed participants. The architecture, elaborated in Figure 14.6, embeds policy configuration, media mixing, and a collaborative virtual environment (CVE) interface within a SIP framework. All the components in this architecture are standard SIP UAs extended with additional user interfaces needed for media policy configuration and control. Communication protocols XCAP (extensible markup language configuration access protocol) and MSCML (media server control markup language) [16] are IETF standards, considered in the following descriptions.

14.4.1 Policy Configuration

In an extended SIP framework, conference participants could configure privacy by sending requests to the policy server using XCAP [15], a standardized way to use HTTP to store, retrieve, and manipulate configuration and application data in XML format. In our proof-of-concept, participants set policies using GUIs to invoke narrowcasting commands on specified controllees, and control is distributed via TCP sockets or HTTP directly (without XCAP).

14.4.2 Policy Evaluation

An application server performs three major functions to evaluate policy:

Evaluating policies configured by each participant: The policy from each participant can be logically compiled into a matrix, as shown in Table 14.4, where entry c_{ij} of the matrix represents connectivity of source i to sink j , and the main diagonal is populated by “don’t care’s”. Each participant (P_1, P_2, \dots, P_n) , where n is the total number of participants, logically sets permissions in authorized cells. Since a media relationship ultimately factors at least two participants, a source and a sink, each cell contains policies from both. For example, $P_1 \rightarrow P_2$, i.e., media sourced at P_1 and sunk at P_2 , has

TABLE 14.4
Policy matrix $P = [p_{ij}]$.

	P_1	P_2	...	P_n
P_1		$P_1(P_1 \rightarrow P_2)$ $P_2(P_1 \rightarrow P_2)$...	$P_1(P_1 \rightarrow P_n)$ $P_n(P_1 \rightarrow P_n)$
P_2	$P_2(P_2 \rightarrow P_1)$ $P_1(P_2 \rightarrow P_1)$...	$P_2(P_2 \rightarrow P_n)$ $P_n(P_2 \rightarrow P_n)$
\vdots	\vdots	\vdots	\ddots	\vdots
P_n	$P_n(P_n \rightarrow P_1)$ $P_1(P_n \rightarrow P_1)$	$P_n(P_n \rightarrow P_2)$ $P_2(P_n \rightarrow P_2)$...	

policy involvement of both P_1 and P_2 : P_1 sets permissions about whether or not to send media to P_2 , and at the same time, P_2 sets permissions about whether or not to receive such media. The policy then is evaluated depending on the combined relationship between P_1 and P_2 .

Responding to participants regarding changes made in the policy:

A policy evaluation report (confirming success or alerting failure of a configuration request) might be sent to participants via standard XCAP response codes.

Sending requests to a media mixer for necessary media mixing: After compiling the media policies, the system determines which media streams need to be mixed and delivered to whom. Using MSCML or some equivalent, the policy server instructs the media mixer to perform the necessary mixing.

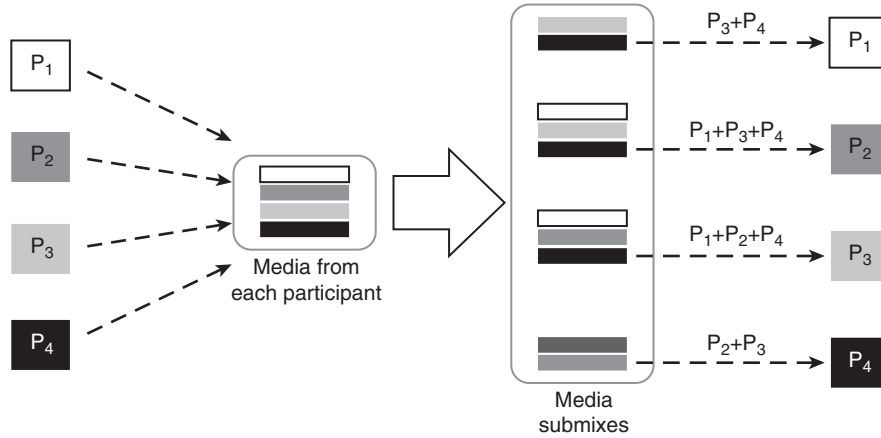
14.4.3 Media Mixing and Distribution

The media server receives MSCML requests from a policy configuration server. According to the accumulated state, it performs the necessary mixing and delivers these streams to subscribed participants. The maximum number of mixes, the power set of the participants excluding the empty and universal sets, is

$$\sum_{i=1}^{n-1} nC_i = 2^n - 2. \quad (14.5)$$

Therefore, for $n = 3, 4, 5$, the maximum number of mixes would be 6, 14, and 30, respectively. However, depending on participants' media privacy requests, the actual number of mixes might be less.

Figure 14.7 illustrates narrowcasting media distribution among four participants when P_1 mutes P_2 and deafens P_4 . All participants send their media

**FIGURE 14.7**

Media mixing and delivery (P₁ mutes P₂ and deafens P₄).

to the media server. The media server mixes only the necessary streams and delivers them back to the appropriate recipients.

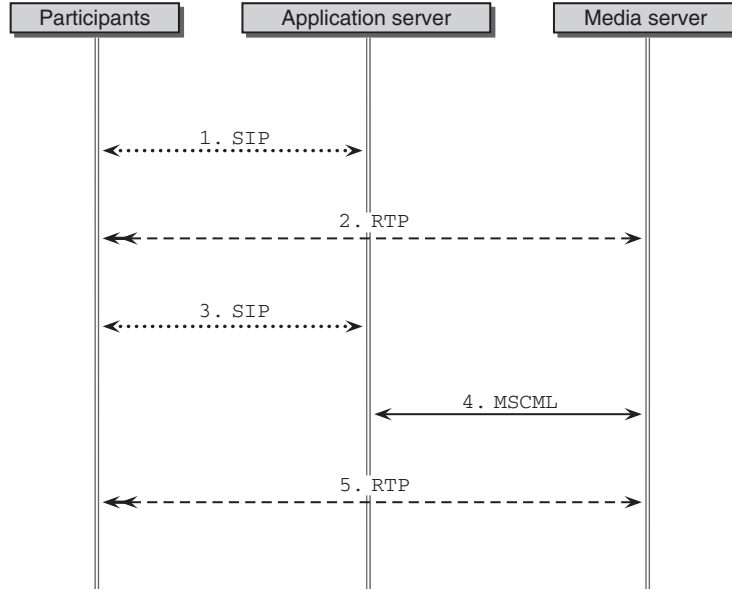
14.4.4 Sample Mixing Configuration

Our prototype environment comprises a SIP server (BEA WebLogic SIP Server), an application server (BEA WebLogic Workshop), a media server (Dialogic/Cantata Snowshore IP Media Server), and four SIP clients (X-lite). We implemented narrowcasting commands `mute`, `deafen`, `attend`, and (partially) `select`, integrating these filter functions into the application server. Figure 14.8 shows the control and media streams among a participant, application server, and media mixer when applying a narrowcasting command.

The MSCML configuration and audibility are shown in Table 14.5 when P₁ deafens P₂. P₁ makes a private group with P₃ and P₄, so P₁, P₃, and P₄ can hear each other, but P₂ cannot hear P₁. The application server evaluates the policy and configures the media server.

14.4.5 Narrowcasting Interfaces

A bridge between CVE clients and SIP-based narrowcasting allows distributed multimodal interfaces [19]. The results of narrowcasting operations are expressed aurally by the SIP-based mixer and visually by graphical interfaces. In contrast to general multimedia systems, virtual environments are characterized by the explicit notion of the position (location and orientation) of the perspective presented to respective users. Often such vantage points are modeled by the standpoints and directions of representative objects in a virtual space. These representatives might be more or less symbolic (abstract icons) or figurative (avatars), but act as delegates of human users. Icons and avatars can reify

**FIGURE 14.8**

Communication flow between SIP entities: A default configuration (1) establishes a normal session (2), but it can be adjusted (3) to reconfigure (4) the mixes returned to the participants (5).

TABLE 14.5

MSCML configuration: P_1 deafens P_2 .

Participant	ID	Team Members	Mixmode	Hears
P_1	P_1	P_3, P_4	Private	$P_2 + P_3 + P_4$
P_2	P_2	None	Full	$P_3 + P_4$
P_3	P_3	P_1, P_4	Full	$P_1 + P_2 + P_4$
P_4	P_4	P_1, P_3	Full	$P_1 + P_2 + P_3$

embodied virtuality, treating abstract presence as user interface objects, symbols and manifestations of sources and sinks. We have developed workstation and mobile interfaces [16] to manipulate narrowcasting attributes in virtual spaces via a Java3D [23] interface. This “virtual reality”-style interface features perspective displays of virtual rooms and spaces with figurative avatars, each of which can be associated with an audio source, corresponding to the voice of a user. A participant can rearrange the locations of avatars in virtual spaces and designate sinks, through whose ears the resulting spatialized soundscape is heard. Also, each participant can apply narrowcasting attributes to the avatars, altering the sound mix. Recalling the monkeys in Figure 14.4, Figure 14.9 illustrates the visual cues used for narrowcasting, including a hand covering the mouth of a muted avatar and

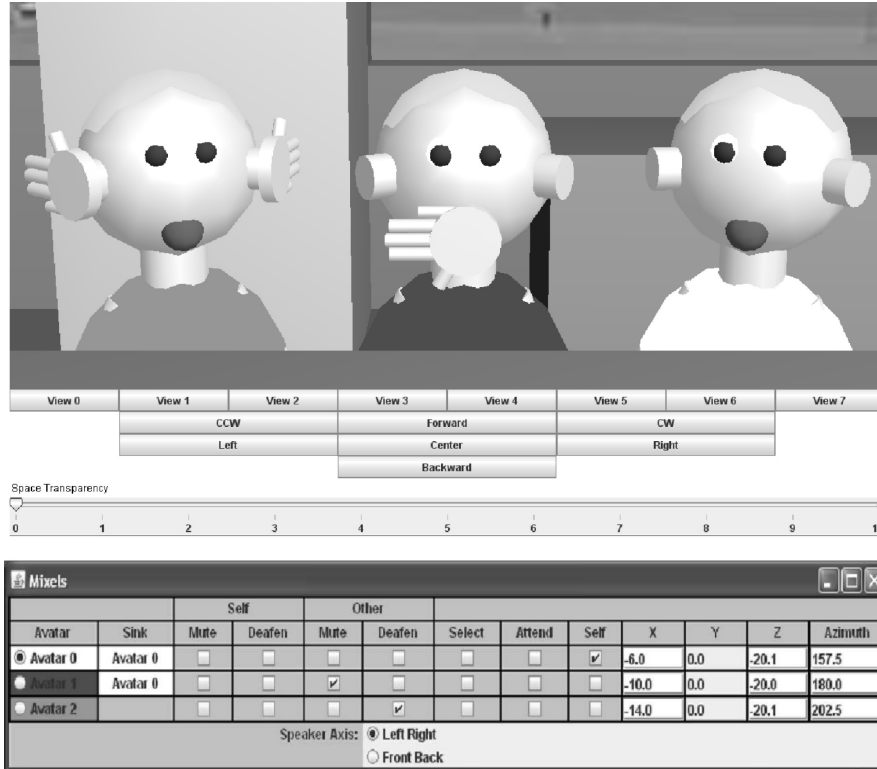


FIGURE 14.9 Narrowcasting Control in Virtual Environment: P_1 (avatar 0, right) mutes P_2 (avatar 1, middle) and deafens P_3 (avatar 2, left).

hands clapped over the ears of a deafened avatar. Manipulations by participants are communicated using a CVE client/server architecture, which framework allows multimodal clients to exchange status data through the network. Clients currently include sound spatializers, telepresence applications, panoramic and turnoramic browsers [10], music visualizers, motion platforms, and mobile interfaces.

Besides the previously-described web- and workstation-based interfaces, we have also built a mobile narrowcasting display and control, shown in Figure 14.10, although we have not yet integrated the mobile audio stream, so it is currently more useful as a collocated “remote control” than as a truly mobile application. Symbolic representations of narrowcasting operations were developed for mobile interfaces by flattening figurative 3D avatars to 2.5D icons, as seen in the second-last row of Table 14.3. In the mobile application, narrowcasting attributes graphical displays are triply encoded—by position (before the “mouth” for mute and select, straddling the “ears” for deafen and attend), symbol (+ for include & - for exclude), and color (green for

**FIGURE 14.10**

Mobile Narrowcasting Interface. Isosceles triangles iconify users. #0 has surged forward and #3 has swayed to the left, but the “ghost icons” show that their local state has not yet been flushed out to the network. #1 has been muted and selected. #2 has been deafened and attended. #3 is in the (currently singleton) selection set, toggled for yawing (spinning).

assert & red, yellow, and orange for inhibit—by self, other, and implicitly, respectively).

The bridge between the interfaces and the SIP-based backend is a ‘read-only’ CVE client embedded in the SIP application server. When the policy server is launched, the embedded client connects to a CVE session server and opens a duplex channel for each member in the conference. Every time a user enables or disables one of the narrowcasting attributes, the action is relayed to the embedded CVE client. As each message is received, the client invokes the necessary methods to reflect the changed status in the SIP conference.

14.4.6 System Performance

The narrowcasting control is basically lightweight: the commands are typically infrequent, and each of them is easily processed by an application server. For excluding narrowcasting commands (*deafen* and *mute*), the time complexity is constant ($O(1)$), independent of the number of participants in a session.

For including narrowcasting commands (`select` and `attend`), in which the connectivity state of the complement of the selection needs to be adjusted, the time complexity is $O(n)$, linear in the number of participants. The configuration for the IP media server used in our experiments supports up to a hundred clients. Even though our laboratory testbed uses a much smaller user pool, typically about four, there is no reason not to assume that the signaling protocol can keep up with practical realtime demands and support the same number of session participants.

14.5 Conclusion and Future Research

The Media Server Component Model architecture can be deployed for policy-based media-mixing and narrowcasting within the standard SIP framework for multimedia conferencing systems. Narrowcasting privacy command interpretation includes a policy evaluation algorithm, a media mixing and delivery mechanism that considers fine-grained policy configured by participants. The policy can be displayed and controlled via interfaces in which hands and other attributes (megaphones and ears trumpets) clapped over figurative avatars' mouths and ears represent audio stream filters. Applications like "Second Life" and virtual environments represent a fertile platform for conferences, distance learning, meetings, and recreational activity like chatspace.

14.5.1 Practical Conferencing

In ordinary conversation, participants generally observe turn-taking, as in a CSMA/CD (carrier sense multiple access/collision detection) protocol with discretionary backup. That is, an utterance that collides with another will cause one or both of the simultaneous speakers to stop and wait until a break before repeating.

One might wonder what happens to such conversational turn-taking in the presence of asymmetric media filters and the absence of a moderator. Narrowcasting features—like blocklists, side channels, and call-within-a-call—complicate teleconferences, since a deafened conversant might not be aware that another is talking and multiple sources might speak at once. If some participants in a conference are muted or deafened to some other participants, without formal floor control there is a likelihood of some "talking on top of" others. In the absence of common floor control, won't private chats and decentralized control lead to cacophonous anarchy? Without "traffic signals," how can collisions be avoided?

In fact, such parallel conversation streams are not a problem. For example, if two participants set up a private side-conference using narrowcasting commands, even though their utterances might collide with others', they wouldn't expect or want others to stop conversing. Rather they "listen with



FIGURE 14.11

Theme-based discussion in articulated chatspace. (© The New Yorker Collection 1961 Robert J. Day from cartoonbank.com. All rights reserved.)

one ear” to ongoing conversations while enjoying their own caucus. Listeners can still untangle conversational threads, by context, voice quality, etc., as suggested by Figure 14.11. Just as in real social situations, including informal gatherings like parties, multiple simultaneous speakers are analyzable. Even “linear” conversations like formal meetings might have some subsets of conversants whispering among themselves while a main speaker is talking. Narrowcasting audio interfaces are even more useful when extended by spatial sound and attenuation based on mutual virtual position (source projection, sink bearing, and distance) [20], distributing the respective voices across a soundscape.

14.5.2 Event Notification Framework for Exchanging Narrowcasting Control Status Information

A narrowcasting conference allows one to influence the media streams of other participants, as well as those of oneself. As a result, each session member can send or receive media streams to and from specific groups in a conference.

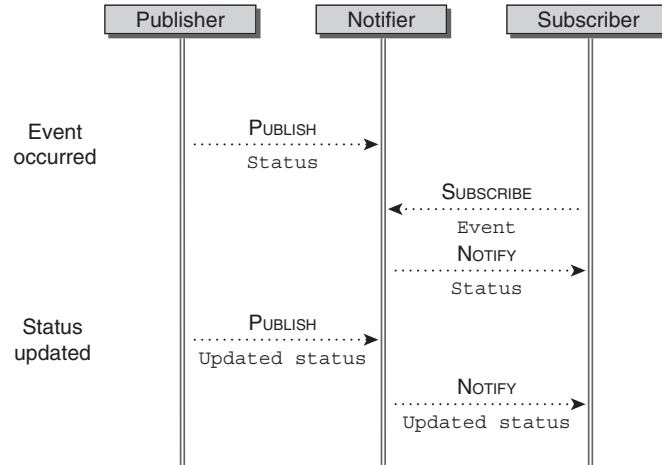


FIGURE 14.12
Event notification mechanism (RFC 3265).

It is potentially useful for participants to be informed of who is controlling media using a narrowcasting command (mute, deafen, select, or attend). For example, teachers may want to be notified whether any students have muted them during a lecture. Depending on the conditions, such role-based transparency could be appropriate or not. A parent might insist upon the ability to override a teenager’s ‘ignore’ filter: “How dare you mute me?” In the current narrowcasting implementation, a notification mechanism has not been incorporated. This section proposes such an event notification mechanism.

The SIP event notification framework (RFC 3265) [22] defines general mechanisms for subscribing to, and receiving notifications of, events within SIP networks. The framework uses standard SIP methods `PUBLISH`, `SUBSCRIBE`, and `NOTIFY` to deliver event-related information. Figure 14.12 shows an event information exchange flow.

In order to allow conference participants to be notified of narrowcasting control status, an event template package can be introduced. A specialized template event package, which might be called “`NARROWCAST`” or “`NC`,” needs to be standardized within the IETF community. Such a template event package could be used in combination with the “`CONFERENCE`” event package (defined in RFC 4575 [23]) to inherit the state of a conference. As shown in Figure 14.13, the publisher (a narrowcasting conference participant), the notifier (presence server), and the subscriber (another narrowcasting conference participant) must support “`CONFERENCE.NC`,” a combination of the `CONFERENCE` event and `NARROWCAST` template event packages. In this example, P_2 subscribes to P_1 and mentions `CONFERENCE.NC` in the “Event” header in the subscription body. As a result, P_2 will be notified of the narrowcasting control information of P_1 .

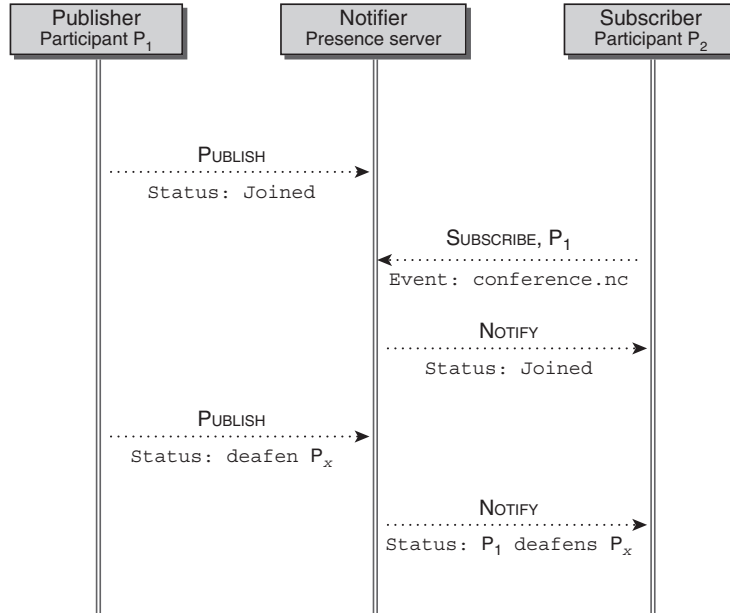


FIGURE 14.13
Narrowcasting control status notification.

14.5.3 Architectural and Interface Refinement

Future research includes allowing selection of multiple sources and sinks for narrowcasting commands to support arbitrary multiuser narrowcasting configurations. A basic challenge is that the SIP side-conference model is not rich enough to support arbitrary multiparty narrowcasting with orthogonal attributes. A related engineering issue is that MSCML is not expressive enough to convey such detailed, articulated channel crossbar control. We are also considering other conference models with multiple policy servers or media mixers. For instance, as multimedia processing becomes less of a specialized service and more of a commodity, a grid computing paradigm could be used instead of a centralized server architecture to mix and deliver media streams to distributed narrowcasting-enabled terminals. `Muffle` (partial deafen) and `muzzle` (partial mute) will enrich the narrowcasting state space [24,25]. We will also generalize policy determination in metasessions with multiple simultaneous chatspaces, in which one has presence across multiple virtual spaces, each with multiple conversants, including “multipresence,” allowing designation of multiple instances of “self” [26].

14.5.4 Convergence

Besides wireline-connected workstation-based interfaces, narrowcasting might find an even more fertile platform in mobile devices. The “four-play” conver-

gence of telephony, television/video, internet, and wireless is driving a proliferation of new devices and services. Mobile terminals, almost as intimate as clothing, are a kind of wearable computer, and a diversity of next-generation functionalities and form factors for smartphones is emerging, including mobile stereotelephony, inspired by cyberspatial audio [27] and augmented audio models. Meanwhile, location-based services—along with seamless handoff, FMC (fixed-mobile convergence), and heterogeneous roaming via MIMO (multiple input/multiple output) smart antennas leading to software-defined radio (SDR) and cognitive radio, leverage geolocation and portable GPS/GIS. Such advanced sensing enables ubicomp and ambient intelligence, including an awareness of user status and availability, and articulated models of privacy, like narrowcasting, which allow users to distribute their attention, availability, and virtual presence. Multipresence and persistent channels, encouraged by ABC (always best connected) networks, will extend the way people communicate.

Acknowledgements

This research has been sponsored in part by grants from Kyushu University Research Superstar Program and the Japan Science and Technology Foundation. We also thank Dialogic/Cantata and BEA for providing media and application servers for our experiments.

References

- [1] Koskelainen, P., H. Schulzrinne, and X. Wu. 2002. A SIP-based conference control framework. In *Proceedings NOSSDAV: 12th Int. Wkshp. on network and operating systems support for digital audio and video* 53–61. New York, NY: ACM Press. ISBN 1-58113-512-2.
- [2] Dommel, H.-P., and J. Garcia-Luna-Aceves. 1995. Floor control for activity coordination in networked multimedia applications. In *Proceedings APCC: 2nd Asian-Pacific conference on communications*. June, Osaka, Japan.
- [3] Even, R., and N. Ismail. 2006. RFC 4825—Conferencing Scenarios. July.
- [4] Spinello, R. A. 2004. *Cyberethics Morality and Law in Cyberspace*. Jones and Bartlett Publishers, Sudbury. ISBN 0-7637-0064-9.
- [5] Berc, L., H. Gajewska, and M. Manasse. 1995. Pssst: Side conversations in the argo telecollaboration system. In *Proceedings UIST: 8th*

- annual ACM symposium on user interface and software technology* 155–56. November, New York, NY: ACM Press. ISBN 0-89791-709-X.
- [6] Yankelovich, N., J. McGinn, M. Wessler, J. Kaplan, J. Provino, and H. Fox. 2005. Private communications in public meetings. In *Proceedings CHI: Human Factors in Computing Systems*, 1873–76. April, Portland, OR: ACM Press. ISBN 1-59593-002-7.
 - [7] Wang, C. C., J. Cahnbley, and J. W. Richardson. 2003. US Patent—Method and system for providing a private conversation channel in a videoconference system. June. Publication No. WO 2003/053034 A1.
 - [8] Fernando, O. N. N., K. Adachi, U. Duminduwardena, M. Kawaguchi, and M. Cohen. 2006. Audio narrowcasting and privacy for multipresent avatars on workstations and mobile phones. *IEICE Transactions on Information and Systems* E89-D(1): 73–87.
 - [9] Alam, M. S., M. Cohen, and A. Ahmed. 2007. Narrowcasting—Controlling media privacy in SIP multimedia conferencing. In *Proceedings IEEE CCNC: 4th Consumer Communications and Networking Conference*. January, Las Vegas.
 - [10] Alam, M. S., M. Cohen, and A. Ahmed. 2007. Narrowcasting: Implementation of privacy control in SIP conferencing. *JVRB: Journal of Virtual Reality and Broadcasting* 4(9).
 - [11] Rosenberg, J., H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. 2002. RFC 3261—SIP: Session Initiation Protocol. June.
 - [12] Rosenberg, J. 2006. RFC 4353—A framework for conferencing with the session initiation protocol. February.
 - [13] Cho, Y.-H., M.-S. Jeong, J.-T. Park, and W.-H. Lee. 2005. Distributed management architecture for multimedia conferencing using SIP. In *Proceedings DFMA: 1st Int. Conf. on Distributed Frameworks for Multimedia Applications*. February, Besancon, France.
 - [14] Cohen, M. 2000. Exclude and include for audio sources and sinks: Analogs of mute & solo are deafen & attend. *Presence: Teleoperators and Virtual Environments* 9(1): 84–96. February. ISSN 1054-7460; www.u-aizu.ac.jp/~mcohen/welcome/publications/ie1.pdf.
 - [15] Rosenberg, J. 2007. RFC 4897—The Extensible Markup Language (XML) Configuration Access Protocol (XCAP). May.
 - [16] Dyke, J. V., E. Burger, and A. Spitzer. 2006. RFC 4722—Media Server Control Markup Language (MSCML) and protocol. November.
 - [17] Walsh, A. E., and D. Gehringer. 2002. *Java 3D API Jump-Start*. Prentice-Hall. Upper Saddle River, NJ, USA. ISBN 0-13-034076-6.

- [18] Sowizral, H., K. Rushforth, and M. Deering. 2000. *The Java 3D API Specification*. 2nd ed. Addison-Wesley. Upper Saddle River, NJ, USA. ISBN 0-201-71041-2.
- [19] Alam, M. S., M. Cohen, and A. Ahmed. 2007. Articulated narrowcasting for privacy and awareness in multimedia conferencing systems and design for implementation within a SIP framework. In *Proceedings ICME: Int. Conf. on Multimedia & Expo*. July, Beijing.
- [20] Cohen, M., and N. Koizumi. 1998. Virtual gain for audio windows. *Presence: Teleoperators and Virtual Environments* 7(1): 53–66. February. issn 1054-7460.
- [21] Cohen, M., and E. M. Wenzel. 1995. The design of multidimensional sound interfaces. In *Virtual Environments and Advanced Interface Design*, ed. W. Barfield and T. A. Furness III, Chapter 8, 291–346. Oxford University Press. New York, New York, USA. ISBN 0-19-507555-2.
- [22] Roach, A. B. 2002. RFC 3265—Session Initiation Protocol (SIP)—specific event notification. June.
- [23] Rosenberg, J., H. Schulzrinne, and O. Levin. 2006. RFC 4575—A Session Initiation Protocol (SIP) event package for conference state. August.
- [24] Cohen, M. 2003. Emerging exotic auditory interfaces. In *114th Convention of the AES*, Amsterdam, March. Preprint # 5819.
- [25] Cohen, M. 1993. Throwing, pitching, and catching sound: Audio windowing models and modes. *IJMMS: The Journal of Person-Computer Interaction* 39(2): 269–304. August. ISSN 0020-7373; www.u-aizu.ac.jp/~mcohen/welcome/publications/tpc.ps.
- [26] Cohen, M. 1998. Quantity of presence: Beyond person, number, and pronouns. In *Cyberworlds*, eds. T. L. Kunii and A. Luciani, Chapter 19, 289–308. Tokyo: Springer-Verlag. ISBN 4-431-70207-5; www.u-aizu.ac.jp/~mcohen/welcome/publications/bi1.pdf.
- [27] Cohen, M., J. Herder, and W. L. Martens. 1999. Cyberspatial audio technology. *Journal of the Acoustical Society of Japan (English)* 20(6). November.

