# Hierarchical Narrowcasting

Michael Cohen(✉)

Spatial Media Group, University of Aizu, Aizu-Wakamatsu,
Fukushima 965-8580, Japan
mcohen@u-aizu.ac.jp

**Abstract.** Narrowcasting, in analogy to uni-, broad-, and multicasting, is a formalization of media control functions that can be used to adjust exposure and receptiveness. Its idioms have been deployed in spatial sound diffusion interfaces, internet telephony, immersive chatspaces, and collaborative music audition systems. Here, we consider its application to desk-top music composition systems, using Pure Data ("Pd"), a dataflow language for audio and multimedia, to develop a proof-of-concept. A hierarchical model of a drum kit is deployed, applying narrowcasting at various levels of aggregation to drum machine sequences. These ideas can also be extended to audio augmented reality situations.

## 1 Introduction

Generally, we are deluged with stimuli, and must devise and deploy strategies for focusing attention or maintaining privacy. There are many techniques for hiding information. For particular instance, hierarchical data structures, presentable as outlines or trees, allow branches to be hidden (collapsed) or revealed (expanded).

Narrowcasting (Cohen, 1998, Cohen, 2000, Cohen and Villegas, 2015)— by way of analogy with broad-, uni-, any-, and multicasting— is an idiom for limiting media streams, formalized by the expressions shown in Fig. 1, to distribute, ration, and control privacy, attention, and presence, as illustrated by Table 1. One could call such deliberate filtering "interinactivity," "disattention," or a "cold spot" (opposite of "hot spot").

"Privacy" has two interpretations, the first association being that of avoiding "leaks" of confidential information, protecting secrets. But a second interpretation means "freedom from disturbance," in the sense of not being bothered by irrelevance or interruption. Narrowcasting operations manage privacy in both senses, filtering duplex information flow through an articulated communication model. Sources and sinks are symmetric duals in virtual spaces, respectively representing media emitters and collectors. Sources can be explicitly "turned off" by muting, or implicitly ignored by deliberately selecting some others. Similarly, sinks can be explicitly deafened or implicitly desensitized if other sinks are "attended."

Previous research explored the power of two-level hierarchies for narrowcasting mixing control. For such arrangements, the terminal tier can be usefully called "sinks," adjustable with `attend` and `deafen` narrowcasting commands, in


N. Streitz and P. Markopoulos (Eds.): DAPI 2015, LNCS 9189, pp. 274–286, 2015.
DOI: 10.1007/978-3-319-20804-6_25

The general, simplified expression of inclusive activation is

$$\texttt{active}(x) = \neg\texttt{exclude}(x) \wedge (\exists\, y\ \texttt{include}(y) \Rightarrow \texttt{include}(x)). \quad (1a)$$

A channel is active unless it has been explicitly disabled or a peer has been focused upon to the exclusion of the respective channel under consideration. So, for `mute` and `select` (or `solo`), the source relation is

$$\texttt{active}(\text{source}_x) = \neg\texttt{mute}(\text{source}_x) \wedge (\exists\, y\ \texttt{select}(\text{source}_y) \Rightarrow \texttt{select}(\text{source}_x)), \quad (1b)$$

`mute` explicitly turning off a source, and `select` disabling the collocated complement of the selection (in the spirit of "anything not mandatory is forbidden"). For `deafen` and `attend`, the sink relation is

$$\texttt{active}(\text{sink}_x) = \neg\texttt{deafen}(\text{sink}_x) \wedge (\exists\, y\ \texttt{attend}(\text{sink}_y) \Rightarrow \texttt{attend}(\text{sink}_x)). \quad (1c)$$

**Fig. 1.** Simplified formalization of two-level narrowcasting and selection functions in predicate calculus notation, where '¬' means "not," '∧' means conjunction (logical "and"), '∃' means "there exists," '⇒' means "implies," and '⇔' means mutual implication (equivalence). The suite of inclusion and exclusion narrowcast commands for sources and sinks are like analogs of burning and dodging (shading) in photographic processing. The duality between source and sink operations is strong, and the semantics are analogous: an object is inclusively enabled by default unless, (a) it explicitly excluded with **`mute`** (for sources) or **`deafen`** (for sinks), or, (b) peers are explicitly included with **`select`** (**`solo`**) (for sources) or **`attend`** (for sinks) when the respective object is not. Because a source or sink is active by default, invoking **`exclude`** and **`include`** operations simultaneously on an object results in its being disabled.

The general expression of two-tier activation is

$$\texttt{active}(object_x) = \neg\texttt{exclude}(object_x) \wedge (\exists\, y\ (\texttt{include}(object_y) \wedge (\texttt{self}(object_y) \Leftrightarrow \texttt{self}(object_x))) \Rightarrow \texttt{include}(object_x)). \quad (2a)$$

A channel is active unless it has been explicitly disabled or a relevant peer has been focused upon to the exclusion of the respective source under consideration. So, for `mute` and `select` (`solo`), the source relation is

$$\texttt{active}(source_x) = \neg\texttt{mute}(source_x) \wedge (\exists\, y\ (\texttt{select}(source_y) \wedge (\texttt{self}(source_y) \Leftrightarrow \texttt{self}(source_x))) \Rightarrow \texttt{select}(source_x)). \quad (2b)$$

For `deafen` and `attend`, the sink relation is

$$\texttt{active}(sink_x) = \neg\texttt{deafen}(sink_x) \wedge (\exists\, y\ \texttt{attend}(sink_y) \wedge (\texttt{self}(sink_y) \Leftrightarrow \texttt{self}(sink_x))) \Rightarrow \texttt{attend}(sink_x)). \quad (2c)$$

**Fig. 2.** Formalization of two-tier narrowcasting and selection functions with **`self`** attribute in predicate calculus notation, where symbols are as above in Fig. 1 and '⇔' means mutual implication (equivalence).

contrast to the start tier labeled "sources," with **`select`** (or **`solo`**) and **`attend`** narrowcasting commands. `Selection` is like plugging headphones into consumer electronic devices that automatically disable loudspeakers: explicitly enabling a sink (the headphones display) implicitly disables other sinks (the speakers display). In graphical applications, the usual "object .. operation" idiom is that enforcedly singleton objects are selected with "radio button" conventions, and that multiple objects can be toggled in and out of the selection set (as with Shift+click). In narrowcasting interfaces, the respective narrowcasting attributes are independent, so "**`solo`**" is rather a misnomer, and "**`select`**" is preferred.

Audio windowing (Cohen and Ludwig, 1991), in analogy to graphical windowing user interfaces, treats soundscapes as articulated elements in a composite display (Begault, 1994). In GUIs (graphical user interfaces), application windows can be rearranged on a desktop, minimized, maximized, and reordered. Audio windowing similarly allows configuration of auditory sources. Soundscapes, analogous to layers in graphical applications, can be combined simply by summing, although in practice some scaling (amplification & attenuation), normalization, equalization, or other conditioning can yield more articulate results.

Having applied narrowcasting to mobile voice communication (Fernando et al., 2009), internet telephony (Alam et al., 2009a, 2009b), and virtual & mixed reality (Ranaweera et al., 2013, Cohen and Villegas, 2015), we now apply it to computer music, deploying it in Pure Data ("Pd") (Matsumura, 2012, Chikashi, 2013), a multimedia dataflow programming environment. There are many digital audio workstation (DAW) and music sequencing & composition applications— such as Audacity,[1] Adobe Audition (formerly Brothersoft Cool Edit),[2] AudioTool,[3] PG Music Band-in-a-Box,[4] Steinberg Cubase,[5] MakeMusic Finale,[6] Apple GarageBand,[7] Abelton Live,[8] Apple Logic,[9] Propellerheads Reason,[10] Avid Sibelius,[11] and SuperCollider[12]— but Pd has an advantage of generality and programability. Almost all such applications feature single-level `mute` and `solo` operations, but only mixing groups resemble anything like multilevel narrowcasting.

By simulating existential quantifiers as list traversal operations (Cohen and Saito, 2008), Pd can implement `mute` and `select` (`solo`) functions on, for instance, a synthesizer. For example, a sample-based drum kit might have its own "sink," collecting the sounds from each of the kick (bass) drum, snare drum, tom-toms, closed and open hi-hats and other cymbals, whereas the bass, rhythm, and lead guitars might be separately miked. However, for nesting depths greater than two, such simple taxonomy is no longer sufficient. With no explicit representation of sinks in such environments, `attend` and `deafen` functions are subsumed by layers of `select` and `mute` controls. A second-level node, which as a sink would have been subject to `deafen` and `attend` for two-level architectures, is a composite source in multilevel topologies (Fig. 3).

## 2 Use Case: Drum Kit

A contemporary drum kit serves as a realistic domain for narrowcasting experiments, since it is multitimbral (different sounds for its components), polyphonic (separable channels so an audio manager can treat atomic sources as layers in

[1] http://audacity.sourceforge.net.
[2] http://creative.adobe.com/products/audition.
[3] http://audiotool.com.
[4] http://www.pgmusic.com.
[5] http://www.steinberg.net/en/products/cubase.
[6] http://www.finalemusic.com.
[7] http://www.apple.com/mac/garageband.
[8] http://www.ableton.com/en/live.
[9] http://www.apple.com/logic-pro.
[10] http://www.propellerheads.jp/products/reason.
[11] http://www.sibelius.com.
[12] http://supercollider.sourceforge.net.

The general expression of multitier activation is

$$\texttt{active}(object_x) \;=\; \neg\texttt{exclude}(object_x) \wedge \\ (\exists\ \mathrm{y}\ (\texttt{include}(object_y) \wedge (\texttt{group}(object_y) = \texttt{group}(object_x))) \Rightarrow \texttt{include}(object_x)). \quad (3a)$$

The function of the `self` flag used above in Figure 2 is replaced by testing for shared membership in groups defined by various `separated` flags. The multilevel relation becomes

$$\texttt{active}(source_x) \;=\; \neg\texttt{mute}(source_x) \wedge \\ (\exists\ \mathrm{y}\ (\texttt{select}(source_y) \wedge (\texttt{group}(source_y) = \texttt{group}(source_x))) \Rightarrow \texttt{select}(source_x)). \quad (3b)$$

**Fig. 3.** Formalization of multilevel narrowcasting and selection functions, where symbols are as above in Fig. 1.

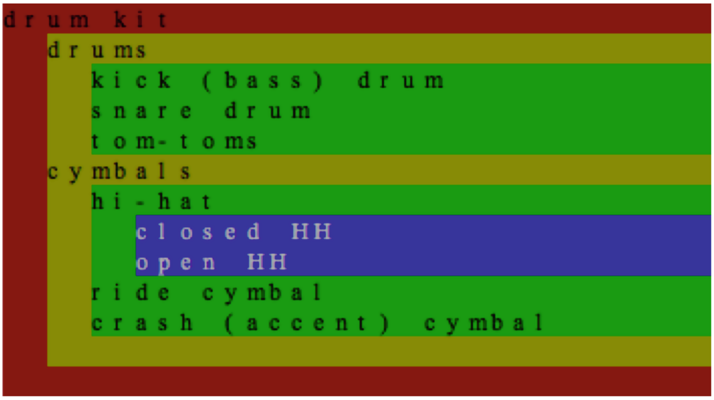



**Fig. 4.** Drum kit used for prototype

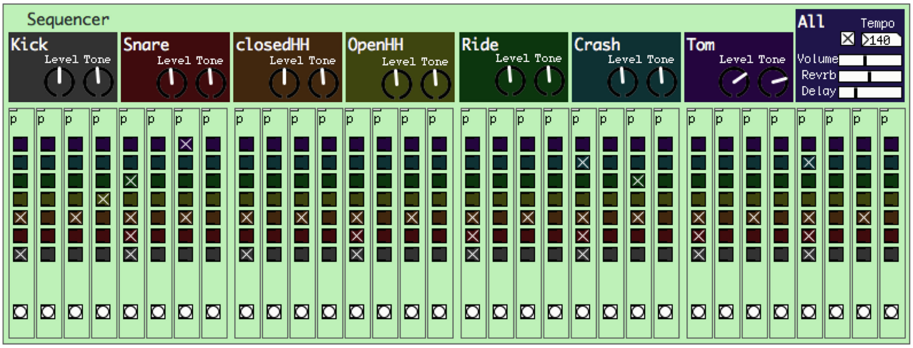


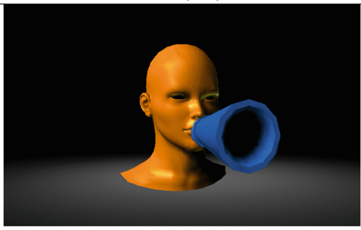
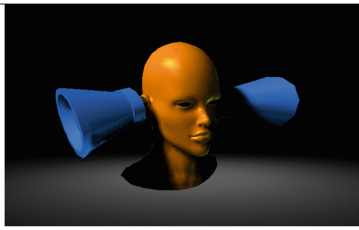
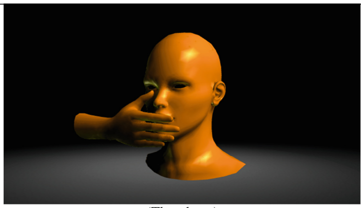
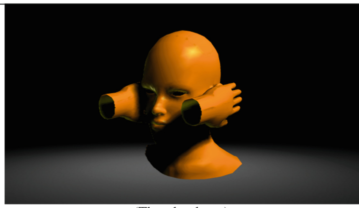
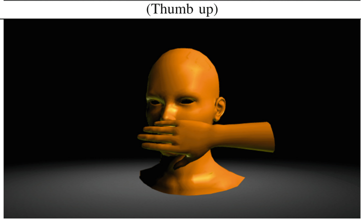
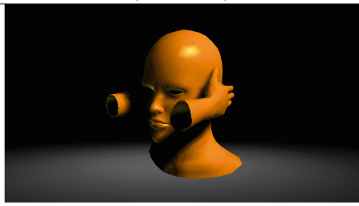

**Fig. 5.** Drum machine sequencer (originally developed by Nobuo Koizumi and Julián Villegas)

a compositable soundscape), and multilevel (modelable as a hierarchical structure). This drum kit is presented in outline form in Fig. 4.

Alternatively and equivalently, such a hierarchy can be expressed as nested lists: $[drumkit[\underline{drums[\underline{kick}, \underline{snare}, \underline{tom}]}, \underline{cymbals[\underline{hi-hat[\underline{closed}, \underline{open}]}, \underline{ride}, \underline{crash}]}]]$.

Such organization is different from taxonomies such as the Hornbostel-Sachs

**Table 1.** Narrowcasting for $^{s}\mathrm{OU}^{\mathrm{rce}}_{\mathrm{Tput}}$ and $^{s}\mathrm{IN}^{\mathrm{k}}_{\mathrm{put}}$. (Figures by Julián Villegas with support of Shun Shiratori.)

| | Source | Sink |
|---|---|---|
| Function | Radiation | Reception |
| Level | Amplification, Attenuation | Sensitivity |
| Direction | OUTput (display) | INput (control) |
| Presence Locus | Nimbus (projection, exposure) | Focus (attention) |
| Instance | Speaker | Listener |
| Transducer | Loudspeaker | Microphone or dummy-head |
| Organ | Mouth | Ear |
| Express | Megaphone | Ear trumpet |
| Include | Select (Solo) | Attend |
| |  |  |
| Suppress | Muzzle | Muffle |
| Exclude | Mute | Deafen |
| own<br>*reflexive* | <br>(Thumb up) | <br>(Thumbs down) |
| other<br>*transitive* | <br>(Thumb down) | <br>(Thumbs up) |

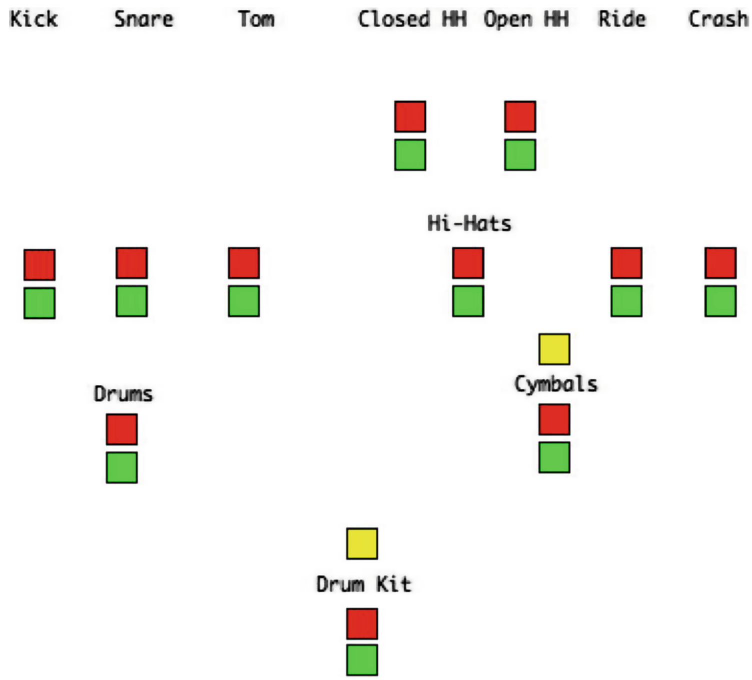



**Fig. 6.** Drum kit narrowcasting interface prototype

classification system since it reflects the practical deployment of the instruments, rather than the "family tree" relation. Henceforth in this description, drum kit components are underlined according to their logical depth in this hierarchical organization.

A drum machine sequencer, originally developed by Nobuo Koizumi and Julián Villegas and the graphical interface of which is shown in Fig. 5, is used for the base of a proof-of-concept prototype, as seen in Fig. 6. The `Interface.pd` module contains a `Sequencer.pd` module, which in turn contains a `Playback.pd` module, which was extended with a newly developed `narrowcasting.pd` controller to enable or disable respective streams. Such a sequencer can be programmed with drum patterns, suggested (Bardet, 1987) or original.

## 3 Multilevel Narrowcasting Hierarchies

A two-level hierarchy is receptive to the source-sink narrowcasting expressions in Fig. 1. In such an organization, `selection` is across all sources so presence or absence of another sink as organizational branch doesn't affect policy. Extension to more than two levels introduces complications, including some considerations that motivate careful definitions of control semantics.

Multilevel `mute` is as straight-forward as its two-level instance: the channel or aggregate is disabled. However, multilevel `select` introduces some subtleties. For instance, should `selection` of a source be considered across an entire hierarchical level of parallel sources, or just in the context of that source's immediate group? For instance, should selecting a cymbal implicitly deactivate a drum? Should `selecting` hi-hat implicitly `mute` other source channels (not only the cymbals, but also the other drums)? They are in different branches of the taxonomical tree, but the usual interpretation of such an operation is to focus on just the selected source, even if there are otherwise active sources in a separate part of the tree. That is, introduction of higher-level organization shouldn't disrupt usual interpretation of such operations. A reasonable case could be argued for either policy. Ultimately, the question is more of a user interface issue than one of semantics.

In two-level teleconferencing scenarios, described by the expressions in Fig. 2, avatars iconifying users have Boolean "`self`" state, indicating association with each respective user. Implicit `mute` due to `selection` is applied only to objects with the same `self` or non-`self` attribute as the object being `selected`. So `selecting` a source unassociated with oneself, such as a remote conversant, would `mute` other conversants but not sources associated with the user making the selection. Likewise, `selecting` one of possibly multiple instances of one's own sources would stifle other `self`-projections, but not affect non-`self` sources.

The same scoping technique can be applied to the example musical application, parameterizing application of implicit `mute` via `selection` by optionally separating related sources. For instance, if a user chose to isolate the hi-hats, `selecting` either open or closed hi-hat would `mute` the other, but not affect other cymbals, let alone the drums. If a user chose not to bundle the hi-hats for the `select` operation, `selecting` one of them would `mute` all the other cymbals.

Likewise, drums and cymbals can also be optionally encapsulated. If, for example, the drums (kick, snare, tom) were so bundled, `selecting` any would `mute` un`selected` drums, but not affect audibility of the cymbals. If the bundling were disabled, `selecting` any drum would have the effect of muting any instrumental source channel, drum or not.

Almost always a human mixer will want at least optional access to the finest (smallest granularity) selection available. That is, one reserves the right to configure a mixing console such that, for example, `selecting` open hi-hat would `mute` the kick drum unless the kick drum were also `selected`.

In the prototype user interface, shown in Fig. 7, `mute` toggles are shaded red and `select` toggles are green, according to usual "traffic signal" conventions. "`Separate`" flags, shown in yellow in the GUI, override the default behavior, asserting separate consideration for the bracket with which it's associated. Any grandparent node can have such an associated `separate` attribute.

Such yellow-colored isolation flags establish a containment. That is, choosing the kick drum will `mute` the ride cymbal unless the drums & cymbals are distinguished, since that separation flag articulates the drum kit, but `selecting` hi-hat containment distinguishes HHs and non-HHs.

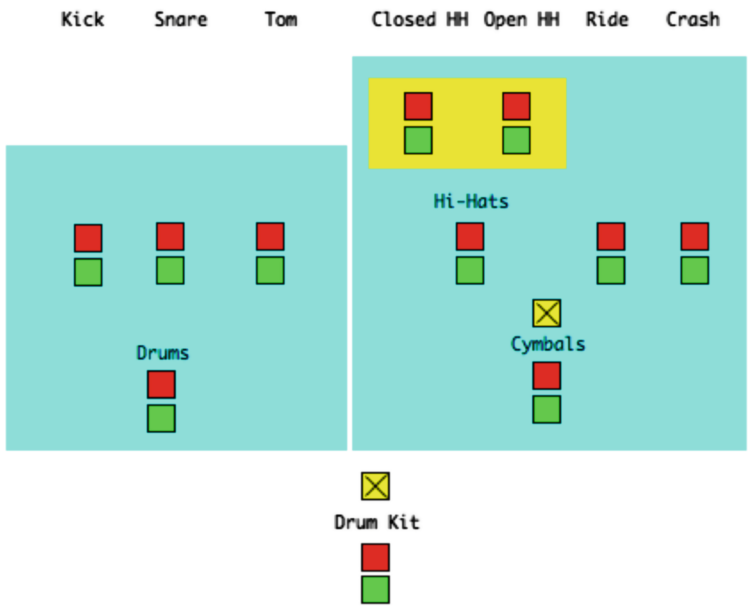



**Fig. 7.** Drum kit narrowcasting interface with separation flags asserted, distinguishing drums vs. cymbals and also hi-hats vs. non-hi-hat cymbals (Color figure online)

If separated, selecting closed or open HH would not affect the other cymbals—ride and crash— but selecting either of those would deactivate the HHs (unless, of course, they were themselves selected, individually or collectively).

A track is active if there is a path on the notional tree from the track's leaf to the root of the hierarchy unimpeded by explicit exclusion or implicit exclusion caused by selection of any unseparated branches. Such atomic track-based modulations are only the beginning of a cascade of stages in the delivery of the composite signal. For audio mixing, the audibility chain also includes other track and mix effects (such as equalization), sequencer intensity, OS gain, and headphone or speaker volume.

As a "thought experiment" (not yet implemented), it's possible that one might, as in Fig. 8, articulate toms = lo ∪ mid ∪ hi. If the toms separate flag were not asserted, lo tom would be active (audible) unless toms (low, mid, & high) were muted, drums (kick, snare, & toms) were muted, drum kit (drums & cymbals) was muted, or mid or hi tom were selected but low tom wasn't, or kick or snare were selected but toms weren't, or cymbal was selected but drums wasn't. If the toms separate flag were asserted, then low tom would be active unless any of its ancestors were muted or mid or hi tom were selected but low tom wasn't. Selection of other tracks outside the separated toms group wouldn't affect activation of any of the toms, including the low tom.

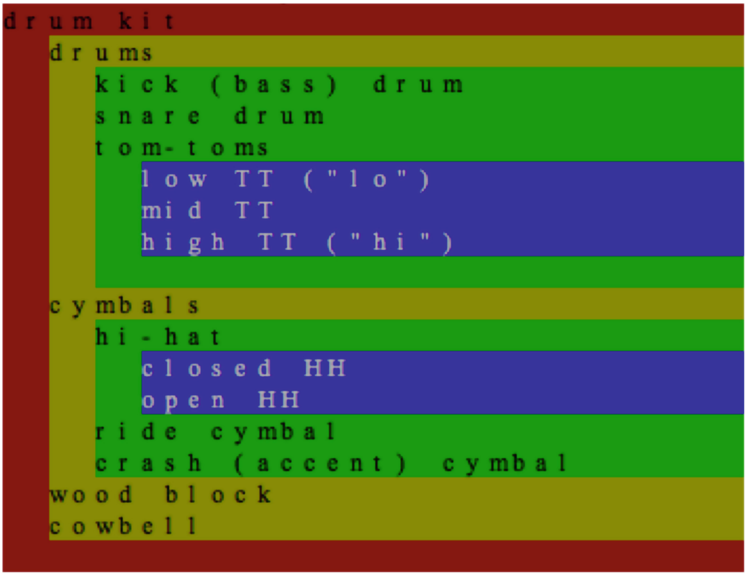



**Fig. 8.** Extended drum kit

If, for another example, another component such as wood block were added to the drum kit (on the same level as the drums and cymbals), then asserting drum kit separation would prevent both the cymbals and the drums from being implicitly `muted` if the wood block were `selected`. Containment in this situation works in both directions: `selection` of an element within such a `separated` group doesn't affect (implicitly `mute`) channels outside the group, and vice versa.

Ultimately such policies are a user interface issue: Mathematical or logical elegance would be no consolation if ordinary operation were awkward or confusing. So the real question is, how will people actually use such controls? The test case explored is not contrived, but small-sized, and proper validation requires a larger domain, such as a 32- or 64-track input situation. The goal is not concision of such narrowcasting expression, but intuitiveness of operation.

## 4 Audio Augmented Reality (AAR) Scenario

Augmented reality (also known, variously, as mixed or enhanced) refers to overlaying synthetic information in displays upon naturally apprehended stimuli. In visual domains, AR composites computer graphics (CG) on top of optical scenes. In auditory domains, AR composites virtual audio signals into natural soundscapes for AAR, audio augmented reality.

As illustrated by Fig. 9, a use-case scenario of an errand-running pedestrian illustrates application of hierarchical narrowcasting to AAR (Cohen and Villegas, 2015). An imagined subject dons an AAR display, *wearware*, perhaps headphones


**Fig. 9.** Hierarchical audio augmented reality soundscape

or nearphones attached to eyewear, or enjoys some blended combination of personal and public display, ubicomp *everyware*. His or her presence is forked, so the subject inhabits multiple scenes simultaneously, *anyware*, with automatic prioritization of soundscape layers, *awareware*. Following a hierarchical taxonomy for virtual perspectives (Cohen, 1998), these layers can be sorted into proximal, medial, and distal classes, described here in decreasing order of intimacy and self-identification, corresponding to utility–professional–recreational classifications.

- **A first-person, utility soundscape:** A proximal "reality browser" layer comprises way-showing, navigation cues: directionalized sounds signifying north, home or origin, intermediate checkpoints, destinations... Populating this layer are auditory icons, recorded everyday sounds with direct associations (such as swooshing of outgoing mail), and earcons, more abstract sounds signifying scene objects (such as ringtone themes associated with particular callers). For instance, the traveler might hear a snippet of a recording of a relative's voice from home, or a tone from a milestone marker. Except for nonliteral distance effects, the soundscape corresponds to a what is sometimes imprecisely called a "PoV" (point-of-view) visual perspective, tightly bound to the position of the listener's head. The rendering is responsive to translation (location) as well as rotation (orientation) of a human user. This intimate perspective can be described as egocentric, centered inside a subject.

- **A second-person, professional soundscape:** At the same time, the user monitors intermittent voicechat among colleagues, whose respective channels are directionalized to suggest arrangement of desks at their office. Relaxing the tight correspondence of a $1^{st}$-person perspective, an endocentric soundscape allows a displaced or telepresent experience, analogous to that often used in games in which players may view their own avatar from over-the-shoulder or behind-the-back. Such medial rendering is sensitive to orientation of the "meatspace" pilot but not location: as a subject physically moves about, the soundscape twists but does not shift. Surrounding voices neither recede nor approach; there is no "parallax." This perspective can be described as egocentric, centered on (but not within) each subject, a kind of metaphorical tethering. Like a $1^{st}$-person soundscape, it is idiocentric, centered upon oneself. Respective sinks are still self-identifiable and personal, but the sensation is more like an out-of-body experience, an "auto-empathic" sensation of being aware that, for example, one's ears are displaced.
- **A third-person, recreational soundscape.** Our subject also enjoys music rendered with respect to his or her head, such as a conventional stereo mix. This distal perspective, perhaps shared with others, can be described as exocentric, centered outside a subject, as it is oblivious or indifferent to the position of the listener. Unlike $1^{st}$- and $2^{nd}$-person aural perspectives, it is allocentric, egalitarian and non-individualized. An entire audience might share a sweet spot.

These layers are combined, basically by adding them, since their perceptual spaces are coextensive, overlaid on the actual environment of the user. Such a composite soundscape is over-crowded, but narrowcasting provides a user interface strategy for data reduction, selectively managing and culling hypermedia. Within each soundscape, sources are individually controllable, either explicitly (via `mute`) or implicitly (by `select`ing others). Narrowcasting sinks fuse multiple layers while preserving their individual activatability (via `deafen` and `attend`), like a "visible/invisible" toggle switch in a graphical application, or an "`ignore`" sender or thread function in e·mail and web-forum browsers. Sinks are designated monitors of respective composited soundscapes; sources are sound effects, distributed voices, and music populating those layers.

## 5 Conclusion and Future Work

Narrowcasting attributes can also be crossed with spatialization and used for "polite calling" or "awareware," reflecting a sensitivity to one's availability, like the "`online`"–"`offline`" status switch of a conferencing service. An incoming call might automatically muzzle or attenuate sources in a music layer by deafening or muffling its associated sink. Focusing on an office conference (by `attend`ing its sink) could stifle other soundscapes. Awareware exemplifies intelligent interfaces that maintain a model of user state, adjusting multimodal I/O to accommodate context and circumstances besides position.

Despite the elegance of the predicate calculus notation, the actual prototype uses explicit expressions to evaluate the active status of the respective tracks. I would like to use such formal expressions as $[\underline{drumkit}[\underline{\underline{drums}}[\underline{\underline{\underline{kick}}}, \underline{\underline{\underline{snare}}}, \underline{\underline{\underline{toms}}}[\underline{\underline{\underline{\underline{lo}}}}, \underline{\underline{\underline{\underline{mid}}}}, \underline{\underline{\underline{\underline{hi}}}}], \underline{\underline{\underline{woodblock}}}], \underline{\underline{cymbals}}[\underline{\underline{\underline{hi-hats}}}[\underline{\underline{\underline{\underline{closed}}}}, \underline{\underline{\underline{\underline{open}}}}], \underline{\underline{\underline{ride}}}, \underline{\underline{\underline{crash}}}], \underline{\underline{cowbell}}]]$ to automatically specify and compile narrowcasting policy. Now, the rules are inelegantly encoded manually, in an ad hoc manner, but it should be possible to completely generate both GUI layout and underlying policy automatically just from such a specification of the connectivity graph.

Modulation of source exposure or sink attention (Benford et al., 1995, Greenhalgh and Benford, 1995) need not be "all or nothing": nimbus (source exposure) and focus (sink sensitivity) can be respectively partially softened with muzzling and muffling (Cohen, 1993). More subtly, frequency-based processing such as LPF (low-pass filtering) can be applied for `muzzle` and `muffle` effects. We plan to extend the Pd prototype to include such signal processing. There are many possible idioms for such controls. For the audio domain used here, such interfaces will be used by sound mixing engineers, not computer scientists, so ease-of-use is more important than mathematical or conceptual flourishes (Cohen et al., 2013).

## References


Alam, M.S., Cohen, M., Villegas, J., Ahmed, A.: Narrowcasting for articulated privacy and attention in SIP audio conferencing. JMM: J. Mob. Multimedia **5**(1), 12–28 (2009a). www.rintonpress.com/journals/jmmonline.html, www.rintonpress.com/xjmm5/jmm-5-1/012-028.pdf

Alam, S., Cohen, M., Villegas, J., Ashir, A.: Narrowcasting in SIP: articulated privacy control. In: Ahson, S.A., Ilyas, M. (eds.) SIP Handbook: Services, Technologies, and Security of Session Initiation Protocol, Chap. 14, pp. 323–345. CRC Press: Taylor and Francis, Boca Raton (2009b). ISBN-10 1–4200-6603-X, ISBN-13 978-1-4200-6603-6, www.crcpress.com/product/isbn/9781420066036

Bardet, R.-P.: 200 Drum Machine Patterns. Hal Leonard Publishing Corporation, USA (1987)

Begault, D.R.: 3-D Sound for Virtual Reality and Multimedia. Academic Press, Cambridge (1994). ISBN 0-12-084735-3

Benford, S., Bowers, J., Fahlén, L., Greenhalgh, C., Mariani, J., Rodde, T.: Networked virtual reality and cooperative work. Presence: Teleoperators Virtual Environ. **4**(4), 364–386 (1995). ISSN 1054–7460

Chikashi, M.: Pure Data Tutorial and Reference, Works Corporation (2013) (in Japanese). ISBN 486267142-X, ISBN-13 978-4862671424

Cohen, M.: Throwing, pitching, and catching sound: audio windowing models and modes. IJMMS: J. Person-Comput. Interac. **39**(2), 269–304 (1993). ISSN 0020–7373, www.u-aizu.ac.jp/mcohen/welcome/publications/tpc.pdf

Cohen, M.: Quantity of presence: Beyond person, number, and pronouns. In: Kunii, T.L., Luciani, A. (eds.) Cyberworlds, Chap. 19, pp. 289–308. Springer, Tokyo (1998). ISBN 4-431-70207-5

Cohen, M.: Exclude and include for audio sources and sinks: analogs of mute & solo are deafen & attend. Presence: Teleoperators Virtual Environ. **9**(1), 84–96 (2000). ISSN 1054–7460, www.mitpressjournals.org/doi/pdf/10.1162/105474600566637

Cohen, M., Haralambous, Y., and Veytsman, B.: The multibibliography package. In: TUG: The 34th Annual Meeting of the TEX Users Group, pp. 83–89 (2013). tug.org/tug2013/, www.ctan.org/pkg/multibibliography

Cohen, M., Ludwig, L.F.: Multidimensional audio window management. In: Greenberg, S. (ed.) Computer Supported Cooperative Work and Groupware, Chap. 10, pp. 193–210. Academic Press, London (1991). ISBN-10 0–12-299220-2

Cohen, M., Saito, H.: Existential quantifiers in Mathematica for narrowcasting predicate calculus. 3D Forum: J. Three Dimensional Images **22**(2), 55–58 (2008). ISSN 1342–2189, jglobal.jst.go.jp/public/20090422/200902289181169720

Cohen, M. and Villegas, J.: Applications of audio augmented reality: wearware, everyware, anyware, & awareware. In: Barfield, W., Caudell, T. (eds.) Fundamentals of Wearable Computers and Augmented Reality, 2nd edn., Chap. 13, pp. 309–330. Lawrence Erlbaum Associates, Mahwah (2015, In press). ISBN 0-8058-2902-4

Fernando, O.N.N., Cohen, M., Dumindawardana, U.C., Kawaguchi, M.: Duplex narrowcasting operations for multipresent groupware avatars on mobile devices. IJWMC: Int. J. of Wireless and Mobile Computing **3**(4), 280–287 (2009). ISSN 1741–1084, 1741–1092, www.inderscience.com/info/inarticle.php?artid=29348

Greenhalgh, C., Benford, S.: Massive: a collaborative virtual environment for teleconferencing. ACM Trans. Comput. Hum. Interac. **2**(3), 239–261 (1995)

Matsumura, S.: Pd Recipe Book– Introduction to Sound Programming with Pure Data. BNN (2012) (in Japanese). ISBN 4861007801

Ranaweera, R., Cohen, M., and Frishkopf, M.: Narrowcasting enabled immersive music browser for folkways world music collection. In: Capin, T., Balcisoy, S., Thalmann, D. (eds.) CASA: Proceedings International Conference on Computer Animation and Social Agents, Istanbul (2013). www.cs.bilkent.edu.tr/~casa2013, ISBN 978-605-4348-53-4